# ISO TC184/SC4/* WG10 N 22 (P 1 )

***Complete with EC (for editing), PMAG, or WG**

**Date: August 11, 1995** **Supersedes SC4/ WG10 N 13 (P 1 )**

**PRODUCT DATA REPRESENTATION AND EXCHANGE**

**Part:** 13 **Title:** Architecture & Methodology Reference Manual

**Purpose of this document as it relates to the target document is:**

x Primary Content
__ Issue Discussion
__ Alternate Proposal
__ Partial Content

Current Status: draft

Version: 4


**ABSTRACT:**

This document is an incomplete draft of the ISO 10303 Architecture & Methodology Reference Manual. Pending a final decision by PMAG, this is designated as ISO 10303-13. The intent of the final version of this document is to provide a complete, definitive, normative statement of the architecture and methodology used in the development of ISO 10303.



**KEYWORDS:**

Architecture
Methodology
Reference Manual
STEP
ISO 10303
Methods
Procedures


**Document Status/Dates**

| Part Documents | | Other SC4 Documents | |
|---|---|---|---|
| 6/10/95 | Working Draft | _____ | Working |
| _____ | Project Draft | _____ | Released |
| _____ | Released Draft | _____ | Confirmed |
| _____ | Technically Complete | _____ | Approved |
| _____ | Editorially Complete | | |
| _____ | ISO Committee Draft | | |


**Project leader:** Julian Fowler
**Address:** CADDETC
6 Blenheim Terrace
Leeds LS2 9HZ
UK
**Telephone:** 44 (0) 113 245 2288
**FAX:** 44 (0) 113 246 8492
**Home tel./fax:** 44 (0) 1943 831688
**E-Mail:** 100255.3453@compuserve.com

**Editor:** Sheila Lewis
**Address:** CADDETC
6 Blenheim Terrace
Leeds LS2 9HZ
UK
**Telephone:** 44 (0) 113 245 2288
**FAX:** 44 (0) 113 246 8492
**Home tel./fax:** 44 (0) 161 976 6874
**E-Mail:** sheila@leva.leeds.ac.uk


**Comments to Reader**

This version of the ISO 10303 Architecture and Methodology Reference Manual is distributed within ISO TC184/SC4/WG10/P1 (formerly WG5/P1) for review and discussion. The document will be reviewed and another version produced prior to the Grenoble meeting of ISO TC184/SC4/WG10/P1 in October 1995. All comments on the document should be submitted to the project leader or editor, before September 22, 1995 to enable the Grenoble version to be produced by September 29, 1995.

this page intentionally blank

# Contents

Notes:

1. This document has been produced following the meeting in Washington DC, June 1995. All editorial requirements documented in the minutes of the Washington DC meeting have been incorporated.

2. In the absence of consensus regarding the structure of this document, the above contents list forms a proposal based on the two sections - architecture and methodology.

3. The methodology section has been based heavily on the Application Protocol Guidelines document (PMAG N103) as this gives an overall picture of the process particularly if the

assumption is taken that Application Interpreted Constructs and Integrated Resources are developed only as required by Application Protocols.

4. Based on the above contents list, the various existing methods documentation has been incorporated into relevant sections for review, comment and update. Identification of gaps and sources for additional material are welcome.

A simple typographical convention is used through-out this document to distinguish the text intended for inclusion in the ISO 10303-13 document, and that included as comments, notes, editorial instructions, or in-line issues. ISO 10303-13 text is presented in accordance with the recommendations and requirements of the ISO/IEC Directives Part 3 and the SC4 Supplementary Directives for ISO 10303. Editorial text, etc., is distinguished by the use of Helvetica font in “boxed” paragraphs (as in this note!).

Within the document, references are included to the members of the ISO 10303-13 project team who have been assigned responsibility for different sections of the document. These references use individuals’ initials, as follows:

| | |
|---|---|
| JPF | Jullian Fowler |
| SPL | Sheila Lewis |
| YY | Yuhwei Yang |
| WCB | Bill Burkett |
| WFD | Bill Danner |
| MEG | Mitch Gilbert |
| DTS | Dave Sanford |

# Foreword

The International Organization for Standardization (ISO) is a world wide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in the subject for which a technical commtttee has been estabvlished has the right to be represented on that committee.

International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

International Standard ISO 10303-13 was prepared by Technical Committee ISO/TC 184 *Industrial automation systems and integration*, Subcommittee SC4 *Industrial data and global manufacturing programming languages*.

ISO 10303 consists of the following parts under the general title *Industrial automation systems and integration - Product data representation and exchange*:

- Part 1 Overview and fundamental principles;
- Part 11 Description methods: The EXPRESS language reference manual;
- Part 12 Description methods: The EXPRESS-I language reference manual;
- Part 21 Implementation methods: Clear text encoding of the exchange structure;
- Part 22 Implementation methods: Standard data access interface;
- Part 31 Conformance testing methodology and framework: General concepts;
- Part 32 Conformance testing methodology and framework: Requirements on testing laboratories and clients;
- Part 41 Integrated generic resources: Fundamentals of product description and support;
- Part 42 Integrated generic resources: Geometric and topological representation;
- Part 43 Integrated generic resources: Representation structures;
- Part 44 Integrated generic resources: Product structure configuration;
- Part 45 Integrated generic resources: Materials;
- Part 46 Integrated generic resources: Visual presentation;
- Part 47 Integrated generic resources: Shape variation tolerances;
- Part 49 Integrated generic resources: Process structure and properties;
- Part 101 Integrated application resources: Draughting;
- Part 104 Integrated application resources: Finite element analysis;

- Part 105 Integrated application resources: Kinematics;
- Part 201 Application protocol: Explicit draughting;
- Part 202 Application protocol: Associative draughting;
- Part 203 Application protocol: Configuration controlled design;
- Part 204 Application protocol: Mechanical design using boundary representation;
- Part 205 Application protocol: Mechanical design using surface representation;
- Part 207 Application protocol: Sheet metal die planning and design;
- Part 210 Application protocol: Printed circuit assembly product design data;
- Part 213 Application protocol: Numerical control process plans for machined parts.

The numbering of the parts of the International Standard reflects its structure:

- Parts 11 and 12 specify the description methods,
- Parts 21 and 22 specify the implementation methods,
- Parts 31 and 32 specify the conformance testing methodology and framework,
- Parts 41 to 49 specify the integrated generic resources,
- Parts 101 to 105 specify the integrated application resources, and
- Parts 201 to 230 specify the application protocols,
- Parts 301 to 330 specify the abstract test suites.

Should further parts be published, they will follow the same numbering pattern.

# Introduction

| |
|---|
| Editorial instruction: Should be aimed at SC4 level.<br><br>SPL: The Supplementary Directives do not give any boilerplate text suitable for this part. I have used the intro. from Part 1 (IS version) here. |

The information generated about a product during its design, manufacture, use, maintenance, and disposal is used for many purposes during that life cycle. The use may involve many computer systems, including some that may be located in different organisations. In order to support such uses, organisations need to be able to represent their product information in a common computer-interpretable form that is required to remain complete and consistent when exchanged among different computer systems.

| |
|---|
| JPF 6/6/95: is this sufficient in terms of the short and long term goals of ISO 10303 and its potential customers in industry? Does this statement exclude the fulfilment of requirements for data management as distinct from data exchange? |

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

| |
|---|
| JPF 6/6/95: the revision of ISO 10303-1 (being undertaken in parallel with the development of this part) may allow change to the last sentence of the previous paragraph. Even though the statement made is (may be) true, it has certainly misled many - including people within the STEP development activity - to expect data sharing and archiving to be achieved using Application Protocols directly. |

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application protocols, abstract test suites, implementation methods and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the description methods series.

This part of ISO 10303 documents the architecture and methodology which supports the development of ISO 10303.

| |
|---|
| JPF 6/6/95: Some of the purposes given below relate to short-term (or SC4 internal) needs. The final version should only cover the longer-term purposes. |

The purposes of this part are to:

- provide the definitive statement of the methods and architecture for approval by SC4;
- present the rationale for methodology;
- be a basis for improvement of the current methods;
- enable the application of the methodology;
- serve as a basis for training in the methodology.

The architecture, methods, and procedures described in this part of ISO 10303 are specified in terms of processes, roles and involvements, not identified working groups, projects or committees. It includes the requirements on project management of the standards development process, as it supports the methodology. It states explicitly the key role of the integration function to the successful use of the methodology. The focus of this part of ISO 10303 is a description of the methodology in terms of its use in developing and enhancing the capabilities of ISO 10303 and other related standards, not a historical perspective.

The target audiences for this document are:

- ISO TC184/SC4 and its constituent national standards bodies and liaisons together with Application Protocol project managers;
- experts in information technology, data modelling, and systems integration[1];
- industry technology experts (Application Protocol teams);
- ISO 10303 developers (those undertaking integration and interpretation work across Application Protocol projects).

JPF 6/6/95: as with the document purposes stated above, some of these audiences are short-term only.

This part is divided into two sections. The first section presents the architecture for ISO 10303, describing its different components and views. The second section describes the methodology used to develop the integrated resources, application protocols and abstract test suite series of parts published within ISO 10303.

## Executive summary

Assigned to: JPF

Editorial notes: Should be able to become free-standing document with minimal editorial change.

Aimed at SC4 level, it will

- provide the definitive statement of the methods and architecture for approval by SC4;
- present the rationale for methodology;
- be a basis for improvement of the current methods;
- enable the application of the methodology;
- serve as a basis for training in the methodology.

(The text of this section is, in fact, based on a separate paper initially delivered at the "STEP Australia" conference in March 1995. There is a set of viewfoils that go with this that are intended for use as introductory training material. Both the paper and the slides are available from JPF on request.)

This part describes the architecture and methods for product data specification developed and used in ISO TC184/SC4; these are the basis for the continuing development of ISO 10303, and are intended to be applicable to the development of other standards and specifications where integration or compatibility with ISO 10303 is required or desired. This part has the dual role of supporting the continued use of the methodology and its review as the basis of future improvement, through evolution.

---

[1] ISO TC184/SC4/WG10 "Technical Architecture" is taken to be representative of this audience, and therefore plays a key role in the development, review and approval of this Reference Manual.

## Industry requirements

The ISO 10303 methodology has been developed to meet industry requirements for standard data specifications that support:

- long term storage and retention of product information;
- the building of bridges between the current “islands of automation” to reduce the number of islands through the integration of systems;

SPL - I think this was an either/or statement, but since no consensus was given on which, I've put both in!

- independence of data from the software tools which create or consume information;
- communication of product information between departments, disciplines, and enterprises.

In addition, the fact that ISO 10303 is a standard introduces additional requirements, in that the specifications developed to fulfil these needs should be stable, extensible, and publicly available.

## Fundamental principles

The ISO 10303 methodology is based on a small number of fundamental principles.

1. ISO 10303 defines an architecture for product data, providing stability and extendibility.
2. ISO 10303 supports and requires traceability of data to product/product context.
3. The role of ISO 10303 is the standardisation of industry application semantics.
4. ISO 10303 defines requirements for the implementation of product data exchange, based on a separation of data specifications from implementation forms.
5. ISO 10303 defines the requirements for the assessment of conformance of implementations.
6. ISO 10303 provides multiple views (projections) of a product over the entire life-cycle.
7. ISO 10303 provides aggregations of characteristics with mutliple representations.
8. ISO 10303 application context is functionally determined.

These principles are the basis for the solutions provided by ISO 10303 to the industry needs articulated above.

## Overview of the ISO 10303 architecture

The architecture of ISO 10303 results from the principles stated above. The complete architecture of ISO 10303 covers all elements of the standard, including the EXPRESS data definition language (ISO 10303-11) and implementation forms such as the ISO 10303 Physical File (ISO 10303-21) and the Standard Data Access Interface (SDAI, ISO 10303-22).

> NOTE – detailed consideration of the architectural aspects of ISO 10303 represented by EXPRESS and the implementation forms is out of scope of this Reference Manual.

The standard data specifications that result from the use of the ISO 10303 architecture and methodology fall into two categories:

- Application Protocols: data specifications that satisfy the specific product data needs of a given industrial application;

- Integrated Resources: generic data specifications that support the consistent development of Application Protocols across many application areas.

This distinction is reflected strongly in the structure of ISO 10303, and its division into several series of parts. Figure 1 provides a high level summary of these elements of the ISO 10303 architecture.

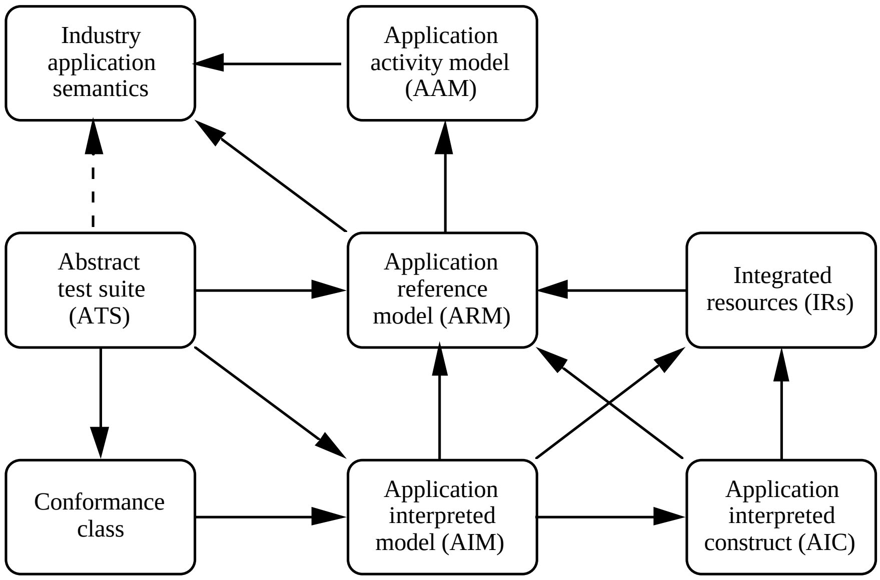


Figure 1: elements of the ISO 10303 architecture

NOTE: The direction of the arrows in the diagram specifies “existence dependence”, i.e., the object at the “tail” of the arrow is dependent on the object at its “head”.

It will be seen from figure 1 that the key element upon which all elements of the architecture are dependent is industrial application semantics. This defines the processes and activities that data is intended to support. For example, ISO 10303 does not support the representation of geometry (points, curves, surfaces, etc.) without linking to a specific purpose in a specific industrial application domain.

The consistency of data specifications within ISO 10303 for a wide range of industry applications (Application Protocols) is ensured by the reuse of common Integrated Resources.

NOTE – the architectural aspects of the ISO 10303 Integrated Resources are described in clause ?? of this part of ISO 10303. The methods used in the development of ISO 10303 Integrated Resources are described in clause ?? of this part of ISO 10303.

The Integrated Resources themselves are based in a formalised framework for product data, sometimes referred to as the Generic Product Data Model (GPDM). This framework defines the basis of all the data specifications that are standardised within ISO 10303.

NOTE – the Generic Product Data Model is described in clause 7.2 of this part of ISO 10303.

NOTE – the original form and documentation of the GPDM is to be found in [4].

# Overview of the ISO 10303 methodology

JPF/SPL 6/6/95: this is obviously incomplete! The overview text from the “Sydney” paper has been distributed across the later sections that deal in more detail with the methodology. Either the original text needs to be moved back here, or new text has to be written.

SPL: 8/2/95: An attempt to rectify the above has now been put in - based on the “Sydney” paper.

The development of STEP is driven by the need to fulfil the diverse needs of many industry sectors in a consistent, cost effective manner. The STEP development methodology therefore not only defines the process by which such needs are fulfilled, but also determines how this process is to be managed.

The STEP methodology is closely focused on the development of Application Protocols i.e. standardised data specifications that satisfy identified industrial needs.

The first stage in articulating this need is the definition, at a high level, of the scope of the proposed Application Protocol and the requirements that it is intended to fulfil. This definition not only enables validation of the proposed Application Protocol by potential users and implementors in industry, but also its assessment for overlaps and redundancies with other Application Protocols.

On the basis of the initial statement of scope and requirements, the proposed AP is balloted as a New Work Item Proposal under ISO rules; if approved, the development of the Application Protocol as a part of STEP is mandated.

The second phase in the development of an Application Protocol is the discovery and documentation of the detailed information requirements that are to be fulfilled. It is important to note that these requirements are discovered, rather than defined: the requirements already exist as the data that underlies industry practices, pro cesses, and systems. These requirements are analysed and documented through the development of an Application Reference Model.

Once the ARM is complete to the satisfaction of the Application Protocol development team, full documentation of the scope (AAM) and requirements (ARM) is distributed as a Committee Draft for Comment (CDC); this process is designed to ensure adequate and effective review and validation of the Application Protocol by experts in industry.

The first two phases in the specification of an Application Protocol are undertaken by the project team responsible for its development. From this point onwards, the further development of the Application Protocol is undertaken through synergy between the project team and the “core” functions of STEP: AIM development, AP integration, resource integration, and ATS development. This interaction may be seen as part of a “matrix” management approach to the development of the standard: each Application Protocol results from the definition of requirements by industry or application experts, the fulfilment of those requirements by “STEP” experts, and the validation of solutions by the industsry experts.

The first of these synergistic phases is the development of the Application Interpreted Model: the creation of a data specificatoin based on the STEP Integrated Resources that meets the requirements stated in the Application Reference Model.

A second part of this analysis, undertaken by the AP Integration team, is the identification of overlaps with other, existing Application Protocols. Where such overlaps correspond to shared requirements across two or more Application Protocols, the development and use of Application Interpreted Constructs, i.e. a shared solution to the common requirements, is enabled.

The third aspect of this analysis is the identification of requirements that are not supported by the STEP Integrated Resources, and therefore give rise to a need for extension to the Integrated Resources. It is an important principle of the STEP methodology that Application Protocols do not themselves define extensions to the resource models. These extensions are developed according to the Resource Integration method.

Following these analyses, the process of application interpretation involves the identification of the mapping from each information requirement (application object, attribute, or relationship) to one or more constructs from the Integrated Resources. This mapping results in the creation of two elements of the documentation of an Application Protocol:

- a Mapping Table, that specifies the precise mapping of each application requirement;
- the Application Interpreted Model (AIM).

The various methods described above relate to the creation of data specifications within ISO 10303. It must be remembered that these specifications are useful only as the basis for implementation of data exchange or sharing, and that such implementations are required to be testable. Requirements for implementation and testing are fulfilled through the specification of conformance classes within an Application Protocol, and of an Abstract Test Suite for the Application Protocol.

The completion of an Application Protocol, Integrated Resource, or Abstract Test Suite initiates the formal processes of review and approval as ISO Committee Drafts (CD) and Draft International Standards (DIS). Responses to comments raised during these reviews gives rise to iterative application of the methods outlined above.

The STEP development methodology governs the development and standardisatoin of data specifications which, when combined with ISO 10303 implementation forms, are suitable for neutral file exchange as well as providing the basis for shared product databases and archiving. The methodology is designed to fulfil a number of high level industry requirements, and is based on a number of fundamental principles that in turn give rise to the architecture of STEP.

# Industrial automation systems and integration – Product data representation and exchange – Part 13 : Description methods: Architecture and methodology reference manual

## 1. Scope

| Assigned to: JPF/SPL |
|---|

This part of ISO 10303 describes the architecture and methodology used in the development of ISO 10303. This description provides not only procedural guidance on the use of the architecture and methodology. It also includes the background, rationale, and theoretical basis for the architecture and methodology.

The following are within scope of this Reference Manual:

- methods for the development of all data standards within SC4;

  NOTE – the intent of the Reference Manual is to enable consistency of standards, and the interoperability of applications, for any standard that follows the methodology.

JPF: By SC4 resolution 75, it is required that P-LIB makes use of this methodology, unless WG2 is able to prove that the methodology cannot be used. Review of initial ISO 13584 documents suggest that the methodology used or assumed in their development differs from that described in this Reference Manual. The development of this document should address this issue, resulting either in the alignment of ISO 13584 development with that of ISO 10303, or by stimulating the development of a corresponding Reference Manual describing the Architecture and Methodology of ISO 13584. In the latter case, ISO TC184/SC4/WG10 will be required to develop and document the higher level "meta-architecture and meta-methodology" that allows the 10303 and 13584 work to be related.

- the methods are those used to develop the data specifications in ISO 10303;
- methods relating to the development of the Abstract Test Suites and Conformance Classes;
- methods relating to Application Protocol implementation strategy;
- requirements for data specification languages, and the usage conventions adopted in applying EXPRESS (ISO 10303-11) in the context of the ISO 10303 architecture and methodology.
- methods relating to the definition of application semantics and their relationship to data specifications;
- methods relating to the interpretation of application semantics within the domain of ISO 10303;
- methods relating to integration over the domain of ISO 10303.

The following are outside the scope of this document:

- the methodology for the development of EXPRESS;
- conformance testing methodology and framework;

  NOTE – the conformance testing methodology and framework of ISO 10303 are described in ISO 10303-31.

- architectures and methodologies associated with implementations of product data exchange, shared product databases, or archiving..

# 2. Normative references

Assigned to: JPF/SPL

This section will be completed as the document matures.

SPL 2/8/95: Which documents become the Normative References?

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

*ISO 10303-1* *Industrial automation systems - Product data representation and exchange: Overview and fundamental principles.*

*ISO 10303-11* *Industrial automation systems - Product data representation and exchange: Description methods: The EXPRESS language reference manual.*

| | |
|---|---|
| *ISO 10303-31* | *Industrial automation systems - Product data representation and exchange: Conformance testing methodology and framework: General concepts.* |
| *ISO 10303-34* | *Industrial automation systems - Product data representation and exchange: Conformance testing methodology and framework: Abstract test methods*[2]. |
| *ISO 10303-41* | *Industrial automation systems - Product data representation and exchange: Integrated generic resources: Fundamentals of product description and support* |
| *ISO 10303-43* | *Industrial automation systems - Product data representation and exchange: Integrated generic resources: Representation structures.* |

# 3. Definitions

NOTE – These will be put into the correct format when agreement is reached on their applicability. The number in square brackets indicates the part in which each term is defined.

**abstract test case (ATC) [31]**: a specification, encapsulating at least one test purpose, that provides the formal basis from which executable test cases are derived. It is independent of both the implementation and the values.

**abstract test group [31]**: a named set of related abstract test cases.

**abstract test method [31]**: the description of how an implementation is to be tested, given at the appropriate level of abstraction to make the description independent of any particular implementation of testing tools or procedures, but with sufficient detail to enable these tools and procedures to be produced.

**abstract test suite [1]:** a part of this International Standard that contains the set of abstract test cases necessary for conformance testing of an implementation of an application protocol.

**application [1]**: a group of one or more processes creating or using product data.

**application activity model (AAM) [1]**: a model that describes an application in terms of its processes and information flows.

**application context [1]**: the environment in which the integrated resources are interpreted to support the use of product data in a specific application.

**application interpreted construct (AIC)**: a logical grouping of interpreted constructs that is shared by two or more application interpreted models.

**application interpreted model (AIM) [1]**: an information model that uses the integrated resources necessary to satisfy the information requirements and constraints of an application reference model, within an application protocol.

**application object [1]**: an atomic element of an application reference model that defines a unique concept of the application and contains attributes specifying the data elements of the object.

[2] To be published.

**application protocol (AP) [1]**: a part of this International Standard that specifies an application interpreted model satisfying the scope and information requirements for a specific application and its relationship to industrial needs.

NOTE – This definition differs from the definition used in open system interconnection (OSI) standards. However, since this International Standard is not intended to be used directly with OSI communications, no confusion should arise.

Danner: issue on definition of Application Protocols: should also include AAM, ARM, and mapping table [issue on Part 1].

**application reference model (ARM) [1]:** an information model that describes the information requirements and constraints of a specific application context.

**application resource [1]:** an integrated resource whose contents are related to a group of application contexts.

**assembly [1]**: a product that is decomposable into a set of components or other assemblies.

**basic alphabet [21]**: the set of characters G(02/00) through G(07/14) of ISO 8859-1.

**clear text encoding [21]**: the encoding of information that only uses 8-bit byte values corresponding to the basic alphabet.

**client (of a testing laboratory) [31]:** the organisation that submits an implementation for conformance testing.

**comparability (of results) [31]:** characteristic of conformance assessment processes such that execution on the same SUT, in different testing laboratories, leads to the same overall summary.

**complex entity data type [11]**: a representation of an entity. A complex entity data type establishes a domain of values defined by the common attributes and constraints of an allowed combination of entity data types within a particular subtype/supertype graph.

**complex entity (data type) instance [11]**: a named unit of data which represents a unit of information within the class defined by an entity. It is a member of the domain established by a complex entity data type.

**component [1]:** a product that is not subject to decomposition from the perspective of a specific application.

**conformance assessment process [31]**: the process of accomplishing the conformance testing activities necessary to determine the conformance of an implementation to an application protocol.

**conformance class [1]**: a subset of an application protocol for which conformance may be claimed.

**conformance requirement [1]:** a precise, text definition of a characteristic required to be present in a conforming implementation.

**conformance testing [31]:** the testing of a candidate product for the existence of specific characteristics required by a standard in order to determine the extent to which that product is a conforming implementation.

**(conformance) test report [31]:** a document written at the end of the conformance assessment process, that provides the overall summary of the conformance of the IUT to the standard for which conformance testing was carried out, and that gives the details of the testing.

**conforming implementation [31]:** an implementation which satisfies the conformance requirements, consistent with the capabilities stated in the PICS.

**constant [11]**: a named unit of data from a specified domain. The value cannot be modified.

**control directive [21]:** a sequence of characters in the basic alphabet.

**data [1]**: a representation of information in a formal manner suitable for communication, interpretation, or processing by human beings or computers.

**data exchange [1]:** the storing, accessing, transferring, and archiving of data.

**data specification language [1]**: a set of rules for defining data and their relationships suitable for communication, interpretation, or processing by computers.

**data type [11]:** a domain of values.

**entity [11]:** a class of information defined by common properties.

**entity data type [11]:** a representation of an entity. An entity data type establishes a domain of values defined by common attributes and constraints.

**entity (data type) instance [11]**: a named unit of data which represents a unit of information within the class defined by an entity. It is a member of the domain established by an entity data type.

**exchange structure [1]**: a computer-interpretable format used for storing, accessing, transferring, and archiving data.

**executable test case [31]:** an instantiation of an abstract test case with values.

**executable test suite [31]:** the set of executable test cases necessary to perform conformance testing of an implementation against a standard or group of standards.

**EXPRESS**: data specification language as defined by ISO 10303-11

**EXPRESS-G**: graphical data specification language as defined by ISO 10303-11

**EXPRESS schema:**

**generic resource [1]:** an integrated resource whose contents are context-independent.

**implementation method [1]:** a part of this International Standard that specifies a technique used by computer systems to exchange product data that is described using the EXPRESS data specification language (ISO 10303-11).

**Implementation Under Test (IUT) [31]:** that part of a product which is to be studied under testing, which should be an implementation of one or more characteristics of the standard(s) based on a given implementation method.

**industrial application context:** the broad set of real world circumstances (e.g., time frame, location, purpose) that constitute an industrial setting in which a thing of interest is done, takes place, exists, or is interpreted that contributes to the direction, rationale, or meaning of the thing.

**industrial application domain:** a synergistic collection of processes (e.g., activities, events), knowledge (e.g., information, constraints, heuristics), and agents (e.g., people, tools, principles, guidelines) within an Industrial Application Context that interact with a purpose to satisfy a need or produce a targeted result.

**industrial application scope:** the range of processes, knowledge, and agents that are implied or entailed by a representation of an Industrial Application Domain (i.e., an Application Domain Model).

**information [1]**: facts, concepts, or instructions.

**information model [1]**: a formal model of a bounded set of facts, concepts or instructions to meet a specified requirement.

**instance [11]**: a named value.

**integrated resource [1]**: a part of this International Standard that defines a group of resource constructs used as the basis for product data.

**interpretation [1]**: the process of adapting a resource construct from the integrated resources to satisfy a requirement of an application protocol. This may involve the addition of restrictions on attributes, the addition of constraints, the addition of relationships among resource constructs and application constructs, or all of the above.

**interpreted construct:** the association of a resource construct with a specific need. It is the atomic element of an AIM or AIC, resulting from interpretation.

**keyword [21**]: a special sequence of characters identifying an entity or a defined type in the exchange structure.

**literal [21]**: an item in a language definition that stands for itself.

**long form:**

**ontology**[3]**:** a set of types which can be applied jointly to classify a domain of discourse.

**partial complex entity data type [11]:** a potential representation of an entity. A partial complex entity data type is a grouping of entity data types within a subtype/supertype graph which may form part or all of a complex entity data type.

**partial complex entity value [11]:** a value of a partial complex entity data type. This has no meaning on its own and must be combined with other partial complex entity values and a name to form a complex entity instance.

**population [11]**: a collection of entity data type instances.

**postprocessor [31]:** a software unit that translates product information from an independent public domain product data format to the internal format of a particular computer system.

**preprocessor [31]**: a software unit that translates product information from the internal format of a particular computer system to an independent public domain product data format.

**presentation [1**]: a recognisable visual representation of product data.

**product [1]:** a thing or substance produced by a natural or artificial process.

**product data [1]:** a representation of information about a product in a formal manner suitable for communication, interpretation, or processing by human beings or by computers.

---

[3] Definition taken from “Technical Report on the Semantic Unification Meta-model. Volume 1: Semantic Unification of Static Models”; ISO TC184/SC4/WG3 N175, October 19, 1992.

**product information [1]**: facts, concepts, or instructions about a product.

**product information model [1]:** an information model which provides an abstract description of facts, concepts and instructions about a product.

**reference path [?]**:

**repeatability (of results) [31]:** characteristic of an abstract test case and derived executable test case(s), such that repeated executions on the same SUT under the same conditions lead to the same test verdict; and, by extension, a characteristic of an abstract test suite and derived executable test suite(s).

**resource construct [1]:** a collection of EXPRESS language entities, types, functions, rules and references that together define a valid description of an aspect of product data.

**section [21]:** a collection of data of the same functional category of information.

**sequential file [21]:** a file that can only be accessed in a sequential manner.

**short form:**

**simple entity (data type) instance [11]:** a named unit of data which represents a unit of information within the class defined by an entity. It is a member of the domain established by a single entity data type.

**statement [1]**: implementation of a given standard. This statement is produced by completing a PICS proforma.

**structure [1]**: a set of interrelated parts of any complex thing, and the relationships between them.

**subtype/supertype graph [11]:** a declared collection of entity data types. The entity data types declared within a subtype/supertype graph are related via the subtype statement. A subtype/supertype graph defines one or more complex entity data types.

**System Under Test (SUT) [31]:** the computer hardware, software and communication network required to support the IUT.

**test campaign [31]:** the process of running the executable test suite for a particular IUT.

**test purpose [31]:** a precise description of an objective which an abstract test case is designed to achieve.

**token [11]:** a non-decomposable lexical element of a language.

**type**[4]: a kind of state of affairs in which objects may participate, i.e., a criterion or set of rules that determine whether an object is of a certain nature, or whether a number of objects are related in a particular way.

**unit of functionality [1]**: a collection of application objects and their relationships that defines one or more concepts within the application context such that removal of any component would render the concepts incomplete or ambiguous.

**value [11]:** a unit of data.

---

[4] Ibid.

**verdict [31]:** a statement of `pass', `fail', or `inconclusive' concerning conformance of an IUT with respect to an executable test case and the abstract test case from which it was derived.

**verdict criteria [31]**: information defined within an abstract test case which enables the testing laboratory to assign a verdict.

## 3.1. Abbreviations used

AAM: Application Activity Model

AIC: Application Interpreted Construct

AIM: Application Interpreted Model

ANSI/SPARC: American National Standards Institute Standards Planning and Requirements Committee.

NOTE – Also used to refer to the three-level architecture for data and data systems developed by the ANSI/X3/SPARC study group who first published the ANSI/SPARC three-layer architecture in 1975.

AP: Application Protocol

ARM: Application Reference Model

B-rep: Boundary representation solid model

CDIM: Context-Driven Integrated Model

CSG: Constructed Solid Geometry

DBM: Database Management

GEDM: Generic Enterprise Data Model

GPDM: Generic Product Data Model

IAD: Industrial Application Domain

IAS: Industry Application Scope

JPF 6/6/95: Industry Application Semantics (term used in clause 4, based on NIST minutes) or Industry Application Scope (used in 7.3)?

ICAM: Integrated Computer-Aided Manufacturing

IDEF1X: ICAM definition language 1, extended

IDEF0: Integration Definition for Function Modelling, or ICAM definition language 0

IGES: Initial Graphics Exchange Specification

IPIM: Integrated Product Information Model

IR: Integrated Resource

IT: Information Technology

NIAM: Nijssen's Information Analysis Method

SC4: Sub-Committee Four

UoF: Unit of Functionality

# 4. Design principles of ISO 10303 architecture and development methodology

This clause:

- identifies the architectural and methodological design principles for ISO 10303 derived from the objectives of the standard;
- specifies the intent and salient characteristics of design components that are solutions to the requirements;
- summarises the development of the architecture and methodology.

| |
|---|
| JPF 6/6/95: the level of detail given in this section is not substantially different from that given in clause 4, or in the introductory text to later clauses. Rather than a structure of introduction – detail – summary being applied to each aspect of the document, we have the same information being presented in varying forms by different authors! It is unlikely that this can be resolved “in committee”: I therefore propose that it should be agreed at the Washington meeting that the project leader and editor be given the authority to undertake major editorial “surgery” to the document to resolve this issue. |

## 4.1. Design principles of ISO 10303

The primary objective of ISO 10303 is to provide a standard data definition to cover unambiguous communication of information that supports the industry functional applications of product life-cycle. These applications are communication of information that are used or created during product design, development, manufacturing or construction, delivery, and maintenance. This objective demanded several design principles of the development of the architecture and methodology for ISO 10303.

| |
|---|
| JPF 5/6/95: is the statement above intended to be a limiting statement of scope with respect to the applicability of STEP? What about operations, procurement, logistics, etc. … |

### 4.1.1. Single integrated structure to facilitate sharing of data

A product data communication standard that intends to serve as the specification to communicate information among multiple unlike application functions, computer systems, and enterprises, demands maximum sharing of data semantics. A design principle was derived that the standard shall be able to abstract common data semantics from multiple disciplines or views into a single integrated structure.

### 4.1.2. Context-dependent semantics

A standard data specification for effective[5] ( i.e., unambiguous) information communication demands a computer interpretable representation. To accomplish this goal, it requires the design of a data

[5] "Effective" should not be confused with "efficient". "Effective" means that the purpose of the communication achieved - that the *meaning* conveyed. "Efficient" means that the communication was achieved quickly, accurately, with a

definition that can precisely and completely state the semantics of data. In order to provide a computer interpretable representation, contrary to popular beliefs, the standard has to be designed with characteristics that are far beyond simply using a language that has formal syntax.

The precision of semantics that is required for effective communication demands semantic specificities that are only understandable and applicable within the intended application context[6]. Thus a design principle was identified that the standard shall be able to specify application-context dependent semantics for an unlimited range of industry functional disciplines.

### 4.1.3. Stability and extensibility

The success of an established international standard depends on the stability of the standard; which encourages industry implementation. The architecture of the standard must also be designed to sustain potential changes and extensions of the requirements. ISO 10303 is intended to be a product data standard that is capable of supporting an unrestricted domain of industries, functional applications within each industry, and life-cycle views, that will expand and change the requirements over time. This characteristic demands a design principle of the architecture to be able to accommodate changes and extensions over time .

| JPF (in NKS mode) 6/6/95: how much of the above can we say in a standard? |
|---|

### 4.1.4. Usability and producibility

With the broad scope and complexity of ISO 10303, the implementation of the entire International Standard by any one organisation, one vendor, or one industry is unlikely. A partitioning scheme is demanded to properly address the usability and producibility of the standard. The partitioning scheme must establish a mechanism for specifying:

- the constructs that are required for information exchange within a defined application context;

- the requirements for conformance testing of implementations of these constructs.

## 4.2. The architecture and methodology of ISO 10303

ISO 10303 has been designed to satisfy these design principles. A technical approach was developed for the design and development of ISO 10303. This approach consists of a comprehensive architecture and a methodology for producing the standard that satisfies the requirements.

The architecture facilitates the development of a complex standard which consists of a number of functional components. Each architectural component has a unique function and is interrelated with other components; together they provide a complete and integrated product. The methodology specifies the procedures and logical work flow to produce a quality and uniform standard.

---

minimum of cost. "Effective" and "efficient" are not always synonymous; in fact, "effective" often requires a solution to be less "efficient".

[6] An application context is the identification of an industry functional application of the data. This identification is used to provide a shared background knowledge that is required for proper inferences regarding the use of data within a field of thoughts or activity. For example, an application context maybe the configuration management function for the design and manufacturing of products or the design and construction information of a building. An application context is defined by the identification of an industry, one or more product disciplines, the product life-cycle stages where the data is relevant in, and the functional activity or application the data is used for.

The data architecture consists of two distinct categories of data specifications: the Integrated Resources (IR) and Application Protocols (AP). The Integrated Resources are generic data constructs that are at a level of abstraction to be applicable to many application contexts. The Integrated Resources are a deep-structurally integrated single model.. They are integrated using a standardized framework (data architecture) for product data.

The Integrated Resources provide a consistent basis for the specification of semantics of all Application Protocols through an interpretation process; This method of development establishes a consistent relationship between the Integrated Resources and all Application Protocols. Because Application Protocols are based on a single integrated model, even if the application contexts vary, all Application Protocols have large “overlapping” data constructs. This architectural characteristic results in application systems that can read the data (although they may not be useful or meaningful) produced according to one AP are able to read data produced by any AP where they overlap.

JPF 6/6/95: if the statement above were true (and accepted to be true), then I doubt that SC4 would have been exercised by the issue of “AP Interoperability” for so long! If this statement is to be retained, it will need considerable justification.

SPL 8/2/95 - problem now resolved?

Application Protocols are data standards which extend, the semantics of Integrated Resources to provide the specificity required for the effective communication under an established application context. Application Protocols are designed to include both the industry requirements and the corresponding ISO 10303 standardised specifications; which represent the usage of the Integrated Resources in the context of the industry requirements. Application Protocols are designed to provide the context-dependent semantics.

The integration method uses a standardised structure as an integration framework. All resource models are extended to address the domain requirements of ARMs and integrated into the framework structure as extensions to the generic semantics of the framework[7]. A specific usage guideline to model all constructs using existence-dependency logic is established as the modelling practice, which provides the extensible quality of the standard. The IRs are standardised components of ISO 10303 provides stability of the standard. All Application Interpreted Models of APs are interpretations as the usage of IRs extends the stability into all existing and future APs. The practice of integration and interpretation involves a centralised single group of people in the methodology ensures consistency and stability in the standard. The integration and interpretation methods together provide the stability and extensibility of the standard.

SPL 6/6/95: is the use of “a single, centralised group of people in integration and interpretation” the best way of fulfilling industry’s needs?

JPF 6/6/95: how does this relate both to the PPC proposals for SC4 reorganisation, and to the proposal (briefly discussed at the Sydney meeting) that this Reference Manual should be an interpretation of ISO 9000 for the development of standard data specifications?

Each Application Protocol is a partitioning that reflects the requirements and usage agreed upon by industrial expertise. Within each Application Protocol, conformance classes are defined to further partition the standard into conformable implementation specifications . Both the Application Protocol and its conformance class partitioning are architectural elements that are designed to provide the usability and producibility.

[7] In other words, new concepts are added to the Integrated Resources by "grafting" or "sewing" them onto the existing structure.

## 4.3. Development of ISO 10303 architecture and methodology

SPL 6/6/95: History! See intro.: “not a historical perspective”. If this material is kept, move to an informative annex.

The development of ISO 10303 architecture and methodology reflects an evolution over time. During the feasibility study (known as the PDES Initiation Study), an architecture of three layers: an application layer, a logical layer, and a physical layer was developed. Many projects developing data models were identified as application layer projects. These projects included mechanical product, curves and surfaces, drafting, tolerances, finite element analysis, etc. The resultant models were published as the ISO 10303 Testing Draft in 1990, called the Integrated Product Information Model (IPIM).

### 4.3.1. The testing draft

IPIM was a large document that collected or interfaced all the data models from the application layer. From the methodological point of view, the IPIM consisted of many models, each of which had its own context and background knowledge. When these models were collected together, interfaces were developed at the obvious touch-points with minimum removal of redundant structures. There was no standardised framework or logical structure used for the construction or integration of these models. From the usage point of view, the strategy for the IPIM, identical to that of IGES was to allow users to identify and select a subset of these data definitions for each communication event.

The publication of IPIM stimulated many comments about its potential short comings: the ambiguity it may introduce, the human interpretation dependency, and the inconsistency in the conformance issues. The question of whether ISO 10303 was an improvement over IGES initiated the proposal of a deep-structure integration of the application area subject models.

### 4.3.2. Integration framework

A highly abstract and generic product data model (GPDM) was developed to define a logical structure for product description and management data. This model was established to be used as the integration framework for the ISO 10303 Integrated Resources — product data subject area. generic concepts. The framework is designed to distinguish the nature of the product data into well defined concepts. These concepts are

- the identification of a product;
- the identification of a collection of characteristics of the product that forms a view of the definition of the product, called “product definition”;
- the identification of a specific characteristic that a product is defined with, called “property definition”; and
- the formal description of the property, called the “representation”.

All Integrated Resource constructs are analysed for their fit into these primary concepts as semantic extensions during the integration process. After integration process, the resulting structure is a single, consistent, deep-structurally integrated model. The objective of creating this deep-structure integrated single model is to capture context-independent meaning (semantics) so that it can provide a baseline for maximum shareability of data. See 7.2 for more details of the Integration Architecture.

### 4.3.3. Integration method

Along with the integration framework proposal was the development of the integration method. The integration method:

1. uses the framework to identify and logically divide the concepts found in individually developed subject area models into the appropriate framework objects. This aspect of the method provides for modularity;

2. extends the semantics of the framework objects and their extensions to incorporate additional concepts; and

3. model with an established set of modelling guidelines and rules to produce a consistent structure.

See “Semantic & Syntactic Rules”, Danner/Sanford/Yang [?].

Prior to the establishment of an integration framework and methods, each application layer model created a self-sufficient “system”. For example, several models defined workable but different solutions of geometry entities and methods for transforming local co-ordinate systems into a global co-ordinate system. During the integration process, models had to be pulled apart and put back together as a part of the Integrated Resources. Common concepts from multiple models are often replaced with a common and often more abstract structure or simply deleted to use an existing construct. As a result, a single integrated model was produced. Because Integrated Resources constitute a single integrated model, a single Integrated Resource Part can no longer contain its own complete system.

### 4.3.4. Partitioning using Application Protocols

ISO 10303 is targeted to support an extensive domain of unambiguous product data communication requirements. This domain includes data necessary to define any product (such as the physical structure of an assembly and the relationship among its components), and the information about the usage of this data for any functional application over the product's entire life-cycle (such as the use of the physical assembly and component structures for providing a manufacturing bills-of-material or engineering parts lists). It was recognised that:

1. no group of industry experts can completely identify all present and possible future application contexts for product data; and

2. no single data model (that has a consistent level of abstraction) can (or should) satisfy the varying and sometimes conflicting application domain specificity (both in data semantics and constraints) that are required for unambiguous communication.

This realisation led to the partitioning of the standard into individual Application Protocols. Each Application Protocol identifies the background necessary for effective communication - the application context. Each Application Protocol establishes the boundary of an implementable and conformable partition of the standard ISO 10303.

### 4.3.5. Interpretation method

Each Application Protocol represents a specific “usage” of the Integrated Resources. The Application Protocol “usage” of the Integrated Resources specifies the scope, constraints, and domain terminologies for the application context. The development method for Application Protocols is designed so that the relationship between Integrated Resources and all Application Protocols are formalised by a mapping process. This mapping process is called the interpretation method. The interpretation method:

1. analyses the Application Protocol requirements (the requirements are represented by application objects, attributes, and assertions);

2. selects the Integrated Resource constructs that provide corresponding semantics;

3. either use them as they are or add constraints for the usage of these selected constructs; and

4. formally documents the mapping in a mapping language.

The Integrated Resource usage constraints specify either:

- the legitimate choice of references in the constructs or

- how legitimate instances of the constructs are populated .

To specify a legitimate choice of references in a construct, EXPRESS rules are written to identify allowable entity types or exclude relationships between SUBTYPE entities. To specify how instances of the construct are populated, attribute values such as string literals or a range of numbers are defined. These constraints are intended to define the conforming characteristics of the Application Protocol. These constraints are not intended to prevent a database from storing instances of the same constructs that are conforming to a different set of rules, caused by incorporating data instances conforming to multiple Application Protocols in the same database.

### 4.3.6. Application Protocol integration method

Although each Application Protocol defines its own boundary for implementation and conformance, it does not result in an independent standard. All ISO 10303 Application Protocols are interrelated. The relationships between Application Protocols exist at two different levels. One level of these relationships is through the use of the Integrated Resources as the common baseline structure. The other level is through the Application Protocol Integration method, to define and include common Application Interpreted Constructs (AICs). Different from the direct sharing of Integrated Resource constructs, Application Interpreted Constructs are identical constructs after the interpretation process. Application Interpreted Constructs contain not only the common baseline Integrated Resource constructs, they also contain application context dependent terminologies and constraints.

The intent for defining Application Interpreted Constructs is twofold: one is to ensure consistent interpretation for identified common requirements found in multiple Application Protocols and the other is to allow for implementation convenience, such that the development of processors or translators may share common codes wherever a common Application Interpreted Construct is used.

Application Interpreted Constructs are designed to provide a logical group of constructs for potential re-use. They are the constructs that identify common usages of the Integrated Resources for multiple Application Protocols. They are however, not intended to identify all shared Integrated Resource constructs among Application Protocols.

### 4.3.7. Conformance definition

Each Application Protocol defines an implementable and conformable standard specification. Within each Application Protocol, the specification may be further partitioned into classes of conformable modules. The intent is to allow for varying levels of computer system’s capability in the same application area to be able to use the same Application Protocol.

# SECTION I - Architecture

## 5. Architectural components

Assigned to: JPF

Editorial instruction: Based on DTS diagrammatic representation of ISO 10303 architecture. Description of this diagram to form basis of chapter.

Editorial instruction: The term “ISO 10303 architecture” to be used to refer to the elements covered in the previous outline as “data architecture” and the “integration architecture” (GPDM and GEDM).

Editorial Instruction (MW): What is a GEDM?

The architecture of ISO 10303 covers all elements of the standard, including the EXPRESS data definition language (ISO 10303-11) and implementation forms such as physical files (ISO 10303-21) and Standard Data Access Interface (ISO 10303-22).

### 5.1. Application Protocols

The ISO 10303 architecture is built to support and facilitate the development of Application Protocols.

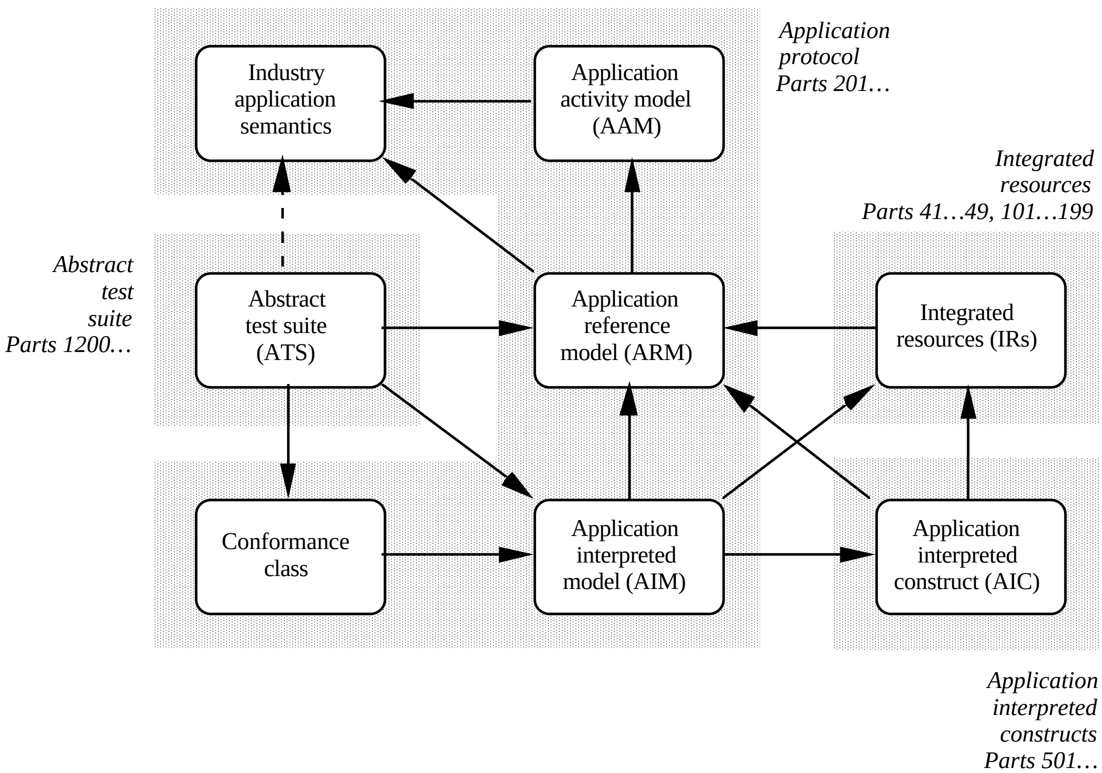


Figure 2: relationship of the ISO 10303 architecture to the documentation of the standard.

Application Protocols employ three types of models, an Application Activity Model (AAM) described in 5.1.1, an Application Reference Model (ARM), described in 5.1.2 and an Application Interpreted Model (AIM) described in 5.1.3.

In addition, the Application Protocol may formally document subsets of the AIM which may be implemented and tested for conformance where the implementor chooses not to implement the whole Application Protocol. These subsets are called Conformance Classes and are described in 5.1.3.1.

### 5.1.1 Industrial application semantics

The Industrial Application Semantics element of the architecture fulfils the principle of traceability of data to industry needs, and also illustrates that ISO 10303 maintains dependence of data on industry applications (what people do) together with independence from computer applications (the tools that are used by people). This linkage to industry applicaton semantics means that all data that complies with ISO 10303 data specifications has an explicit, independent link with the reason or purpose for its existence. For example, ISO 10303 does not support the representation of geometry (points, curves, surfaces etc.) without linking such a representation to a specific product, discipline, life-cycle phsae etc. Therefore complete information about the shape of a product can be changed between organisations without the need for additional communication by phone, fax etc., to indicate the purpose of the exchange.

### 5.1.2 Application Activity Model

Industry application semantics are defined by reference to an Application Activity Model (AAM); this model, describes the activities and processes that use and produce product data in a specific applicaton context. The AAM is defined in IDEF0 [5], a formal process modelling language. It supports the analysis of the activities and information flows within the scope of the industry application. Further detailed analysis and design of data specifications within ISO 10303 is linked back to the “in scope” activities and information flows. It should be noted that the role of the application activity model is to capture the activities ("what is done") within an industry application, not the detailed processes ("how it is done"), that are likely to vary between organisations, or with time as the result of continuous improvement or business process re-engineering activities.

### 5.1.3 Application Reference Model

The Application Reference Model results from a detailed analysis of the requirements of the industrial application. It is a detailed specification of the application objects (entities and attributes), and the relationships between them, that are required to support the activities within the scope of the industry application. The Application Reference Model is documented in a formalized modelling language such as EXPRESS, IDEF1X, or NIAM with each information requirement having a normative definition.

This specification is prepared through analysis of requirements identified by experts in the industry application (sometimes referred to as “domain experts"). These requirements are therefore described using the terminology of the application, and form the basis not only for further development, but also for review and validation. As figure 1 illustrates, the Application Reference Model is dependent on the Application Activity Model: it is a detailed description of the data that supports and flows between the activities described in the Application Activity Model.

JPF 6/6/95: this reference is to the diagram included in the Executive Summary. Is this of sufficient detail? Should the diagram be repeated here for ease of reference?

SPL 8.3.95: YES - repeat the diagram

Should the more detailed diagram produced by Dave Sanford after the NIST workshop, and/or JPF’s EXPRESS-G representation of the detailed structure of the ISO 10303 architecture be used here instead?

Editorial instruction: Techniques for ARM development required or cite to document.

### 5.1.4. Application Interpreted Model

The Application Interpreted Model is a model of selected integrated resources which are constrained, specialized or completed to satisfy the information requirements of the Application Reference Model. The Applicaton Interpreted Model is defined in EXPRESS and EXPRESS-G (a graphical subset of EXPRESS).

This re-use of standard constructs across a wide range of industry requirements results in a high degree of consistency and integration across models, and enables potential reuse of the software code used in interfaces and the potential sharing of common data across application domains. The Application Interpreted Model specifies the data constructs to be used in achieving exchange of information between computer applications; because the Application Interpreted Model is defined using the EXPRESS language, and therefore enables file-based exchange in conjunction with ISO 10303-21, or data access using ISO 10303-22.

#### 5.1.4.1 Conformance Classes

An Application Interpreted Model, as described above, provides the normative specification for data to be exchanged between computer applications. This provides the scope and boundaries for implementations of product data exchange that conform to ISO 10303, and also the scope and boundaries for testing implementations. In order to meet the needs of differing computer systems used within a given industrial application, whilst maintaining consistency of implementation and testing, two or more Conformance Classes may be defined for an Application Interpreted Model.

> NOTE – if no conformance classes are defined for an Application Interpreted Model, it is required that conforming implementations implement the complete Application Interpreted Model, i.e., that the full Application Interpreted Model is the sole conformance class defined.

A conformance class defines a subset of the Application Interpreted Model that may be used as the basis for implementation and testing. These subsets define the minimum conforming implementation based on the Application Interpreted Model; implementations based on any other subsets are not considered to be conforming.

### 5.1.5. Mapping Tables

The interpretation process requires that the constructs in the Application Reference Model are mapped to suitable constructs in the Integrated Resources. This process results in two sections of the Application Protocol, the Application Interpreted Model and the Mapping table.

The mapping table is documented using the mapping language. The mapping table does more than simply document the relationship between constructs in the Integrated Resource and application objects from the Application Reference Model. The mapping table provides additional information (e.g., reference paths, mapping rules) required to create, import, or interpret application objects into an application system. An application system that can read in a data file based on an Application Interpreted Model long form is necessary, though not sufficient for using an Application Protocol. The mapping table provides the additional rules for transforming the data into objects that the application understands.

## 5.2. Application Interpreted Constructs

Another key element of the ISO 10303 architecture is the Application Interpreted Construct (AIC). The process of interpretation within the ISO 10303 methodology is the selection and possible constraint of integrated resource constructs to meet an identified industry need. When a common requirement is identified across two or more industry applications, an Application Interpreted Construct may be documented as being the shared fulfilment of this requirement. An Application

Interpreted Construct explicitly identifies the potential for shared data between industry applications. Application Interpreted Constructs are specified using EXPRESS.

## 5.3. EXPRESS

### 5.3.1. Model specifications

Three elements of the ISO 10303 Architecture: Application Interpreted Models, Integrated Resources, and Application Interpreted Constructs, are specified using the EXPRESS language. The complete specification of each of these elements has, in fact, two elements: the data specification (in EXPRESS), and the specification of the meaning of the data (in English). The structures of the EXPRESS language themselves, of course, provide a partial definition of the meaning of the data, but the complete, unambiguous semantics are conveyed only by the combination of both specifications. To aid understanding of the structure and inter-relationships of these models, a third element is included in their documentation: a graphical presentation of the data specification using the EXPRESS-G notation. EXPRESS-G is documented in ISO 10303-11, Annex D.

## 5.4. Abstract Test Suites

The importance of testing and testability within ISO 10303 is reflected by a standardised framework and methodology for conformance testing. An abstract test suite (ATS) specifies, in non-specific or parameterised form, the abstract test cases from which executable test cases may be derived and used in assessing the conformance of an implementation to the data specification contained in an Application Interpreted Model and the other elements of the ISO 10303 architecture upon which an Application Interpreted Model depends. Experience in other domains, such as the OSI standards for Open Systems, has shown that standardisation of Abstract Test Suites is an essential prerequisite to repeatability and consistency of testing, and therefore of mutual recognition of test results across regional or national boundaries.

## 5.5. Integrated resources

The standard data constructs used in the creation of an Application Interpreted Model are specified in context independent models: Integrated Resources (IRs). These are data models that reflect and support the common requirements of many different product data application areas. The Integrated Resources logically constitute a single, conceptual model for product data. Although the Integrated Resources are used as the basis for developing Application Interpreted Models, they are not themselves intended for direct implementation: they define reusable components that are intended to be combined and refined to meet a specific need. Integrated Resources are specified using EXPRESS.

## 5.6 Implementation methods

### 5.6.1 ISO 10303-21

### 5.6.2 ISO 10303-22

# 6. Data architecture

| Assigned to: WCB |
|---|

## 6.1. Data specifications

Assigned to: WFD

Editorial instructions: include Industrial Application Scope (IAS), Application Activity Model, Application Reference Model, Integrated Resource, Application Interpreted Model, Application Interpreted Construct

Include updated N130?

### 6.1.1. The ISO 10303 integration framework and the role of application protocols

Text extracted from Application Protocol Guidelines section 1.2 on integrated resources.

The ISO 10303 integration framework establishes an explicit architecture for the conceptual models that are part of ISO 10303. This architecture provides the structure for the integrated resources and application protocols. The integrated resources provide constructs that are independent of a specific product data application context. These constructs are used for developing the application interpreted models of application protocols.

### 6.1.2. Application protocol models

Application Protocol Guidelines section 1.2.1 on application protocols

SPL 6/6/95: this repeats/supplements information already given in clauses 6.1, 6.2, and 6.3.

Editorial instruction: includes Application Interpreted Constructs

## 6.2. Standard data constructs

Assigned to: YY/WFD/JPF

Editorial instruction: include: application context, product definition, product property definition, product property representation, representation, presentation.

Previously integration architecture section. To present concepts underlying the GPDM and GEDM. The actual data constructs standardised in the current step irs to be used to illustrate the concepts.

Within ISO 10303, elements of data specifications (or “constructs") are taken to be the representation of facts about objects in the real world. The basis of ISO 10303 data specifications lies in a framework for product data modelling that is based on a classification of the types of data that describe products. This classification identifies five major types of data, as follows.

- **Application context:** data that defines the purpose for which product information is created, and the types of product, disciplines, and life-cycle stages for which such information is valid. The use of an application context allows data that represents an “as designed” product to be distinguished from that for an “as built” configuration, etc.

- **Product definition:** data that identifies products, including variants and categories, and that defines life-cycle “views” of products. Product definition data also includes that which relates to the structure of products, in terms of assembly structures, configurations, effectivities, bills of materials, etc.

- **Product property definition:** data that characterises products by their properties, independent of the representation of properties. For example, it is possible to identify the shape of an object, or aspects of the shape, as a property of the object, without providing a detailed description of shape using a CAD model, engineering drawing, etc.

- **Product property representation:** data that represents the properties of a product, including multiple representations of the same property. For example, the shape of an object may be identified, and then described in many different ways: a 3D CAD model, a physical mock-up, an engineering drawing, and a technical illustration are different representations of the same shape.

- **Product property presentation:** data that defines the presentation of product information to support human communication. The shape of an object (the property) is represented by co-ordinate values, curves, surfaces, etc.; this representation is presented by assigning colours, line fonts, etc. and displaying the resulting picture on a workstation.

This classification of product data is the basis for all ISO 10303 data specifications. It is the framework upon which all the Integrated Resources are built, and is reflected in the Application Interpreted Models (AIMs) of all Application Protocols. The models that capture this framework embody the principle of existence dependence, which ensures that all product information is related to an identified product and ultimately to an application context. This structure is summarised graphically in figure 2 below.

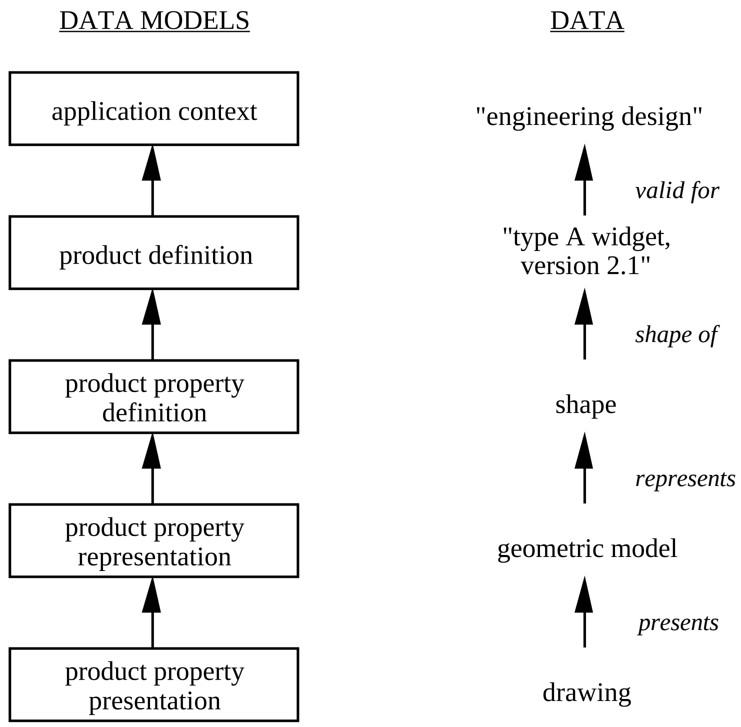


Figure 3: existence dependence of ISO 10303 models

NOTE – This principle can lead to models that are at first sight counter-intuitive: rather than stating that a product has a shape, a shape is “of” a product. However, simple analysis of this example shows that the existence dependent form of the model requires that a shape is always the shape of a product.

Similarly, at a lower level in the structure, ISO 10303 does not allow the existence of "geometry" data as collections of points, lines, curves, etc. Through the existence dependent structures in the ISO 10303 models, such a collection of geometry data must be the representation of some property, that is related to the definition of a product, that has validity in some application context. Thus the basic structure of the ISO 10303 models satisfies and enforces the principle identified above that all product data shall be traceable to an industry need.

JPF 6/6/95: just as we have a high level overview of the GPDM, there should be an equivalent sub-clause that discusses the role of the GEDM (and therefore of the management resources) to the STEP data specification architecture.

## 6.3. Application Contexts

WCB: Assumptions that I assume are explained somewhere else in the document.

1. Application Interpreted Models do not expand the "vocabulary" constituted by the integrated resources. Only specialisation's and constraints are added to an Application Interpreted Model.

2. The purpose of the Integrated Resource constructs is to have a limited set of constructs (vocabulary) that can be ascribed different semantics depending on context. This is intended to resolve the semantic richness/limited entity set dichotomy.

No communication takes place without some kind of context. The context of a communication provides additional information needed to interpret and ascribe meaning (i.e., understand) to transmitted symbols. This is as true for communication between automated systems as it is for communication between people.

The concept of an Application Protocol is innovative and unique among information technology standards because it explicitly uses the context as an operational aspect of the standard. The combination of the Scope clause, the Application Activity Model (AAM), and Application Reference Model (ARM) establishes the scope of an Application Protocol within the context of product data use.

It is important to note that the Application Reference Model not only contributes to the definition of the scope of an Application Protocol, but also represents the *information requirements* that must be met by the Application Protocol. The Scope clause, Application Activity Model, and Application Reference Model layout and describe the boundaries of the application domain; the Application Reference Model serves as a specification the information requirements (of interest) within that domain.

The purpose of this section is to explain the role and impact of Application Contexts with respect to Application Protocol development. This purpose shall be achieved by presenting:

- the role of Application Context with respect to the Application Protocol;
- approaches (e.g., taxonomies) for classifying and characterising an Application Context;
- how Application Contexts - and thus Application Protocols - relate to one another;
- the rationale for the way in which the ISO 10303 Development methodology approaches the definition of Application Contexts.

### 6.3.1. Genesis of the needs and use of Context in Data Exchange

SPL 6/6/95: historical – move to Annex ??

For several years prior to incorporation of Application Protocols into ISO 10303, the IGES community had been working on IGES Application Protocols. IGES Application Protocols were developed in response to IGES “flavours”; they documented a subset of the IGES entity set and specified rules for using the entities correctly within a specific application domain.

Similarly, Test/evaluation teams in the first years of PDES, Inc., realised that they could not evaluate ISO 10303 without recourse to an explanation of the usage of the standard. They developed a Context-Driven Integrated Model (CDIM) to provide a usage context for testing and evaluating ISO 10303 schemas. A report on the activities that led to the development of CDIMs stated:

> “… Testing of these models under the same application context will yield validation of the usefulness of these models …
>
> “A context driven integrated model is an integrated model which is composed by data definitions and data construct relationships, constraints, etc., … that can provide data requirements for a specific application context … CDIMs will be the bridge between application user's requirements and the conceptual models …”

Thus, the need to test the standard required the definition of an application context within which to conduct the testing.

While the reasons that led to their development differed, the concepts inherent in Application Subsets and CDIMs were sufficiently alike to give birth to a new element of ISO 10303 - the Application Protocol. One of the purpose of an Application Protocol and its structure is to allow the usefulness and correctness of the data specification to be reviewed and validated without relying on “field testing” through actual industrial use of the standard (as was the case with IGES).

### 6.3.2. Application Domains, Contexts, and Scopes

In order to present abstract concepts with more semantic precision, some terms need to be defined as they are used through this section. The terms are:

- Industrial Application Context;: this is specified by the entire Application Activity Model, i.e.., it includes the “out of scope” activities and flows;
- Industrial Application Domain;
- Industrial Application Scope: selected elements of Application Activity Model for the Applicatoin Protocol.

During the development of the initial release of ISO 10303, these terms and variations thereof where used interchangeably and with variable shades of meaning. The definitions provided in clause 3 for “Industrial Application Context”, “Industrial Application Domain” and “Industrial Application Scope” assign a specific term to a specific meaning and thereby give a name to concepts used in discussion during the development of ISO 10303.

An Industrial Application Context is a human abstraction of the totality of details within some realm of action, time, or existence that is an acknowledgement of the influence of the details on one another. An Industrial Application Context can only be described in broad brush terms, yet it is an understanding essential to the meaning of the details.

Because it exists in the real world, an Application Domain is fuzzy - it is not clearly bounded or delineated. Language and written symbols can be used to *represent* the Application Domain. Depending on the viewpoint and purpose of identifying and referring to it, an Industrial Application Domain may be very large or very small; examples of Industrial Application Domains are :

- design of a space station;
- finite element analysis of the structural behaviour of a beam;
- production of a cost estimate for the fabrication and construction of a chemical plant;
- replacement of a defective circuit board assembly;
- approval of a design change;
- decommissioning of a nuclear energy facility.

As is true for **all** representations of real-world phenomenon, any representation of an Industrial Application Domain is biased, incomplete, and approximate. There are an infinite number of things to know about any given real-world phenomenon (i.e., Application Domain or thing, process, constraint, etc., within the Application Domain); a representation selects a subset of those things that are important to the purpose of the representation[8].

It is the representation of an Industrial Application Domain that establishes the scope of an Application Protocol. This representation consists of the Scope, Application Activity Model, and Application Reference Model of an Application Protocol; collectively, they may be referred to as an Application Domain Model.

An Application Domain Model describes, documents, bounds, and delineates the Application Domain for the purposes of establishing information requirements for an Application Protocol. The model should be such that one can determine whether a given action takes place within the Industrial Application Domain, a piece of information is used within the Industrial Application Domain, or an individual or mechanism participates in the Industrial Applicatoin Domain. The application *scope* is the range of things thus entailed.

The Application Domain Model serves as a *specification* of the scope of an Application Protocol. Therefore, when one discusses an Industrial Application Scope it is in reference to the things encompassed or entailed by the specification.

### 6.3.3. Application Context Taxonomies

The notion of a context for the use of product data is easy to understand because people recognise the use and role of context in everyday conversation. Describing the context, however, is much more difficult. What exactly are the salient features of the “circumstances in which a particular event occurs” and how can they be described?

Since the introduction of the Application Protocol concept, there have been several attempts to define taxonomies, ontologies, and classification structures for the description, analysis, and planning of the scopes of Application Protocols.

#### 6.3.3.1. Use of taxonomies

The most common thread through the extant taxonomies, frameworks, and classifications is the Lifecycle categorisation. This is followed by categorisation based on the empirical typing (kinds) of product data; and by then by industry sectors or product type.

Taxonomies have not been widely used in the planning and development of Application Protocols for a number of reasons. The first and most obvious is that the classification schemes are not well-

[8] The representation is in this sense a *model* of the real-world phenomenon.

structured; if Lifecycle, Product Data type, and product type were treated as axes, they would not be orthogonal to one another.

A further, and perhaps more cogent, reason is that most uses of product data cannot easily be pigeonholed in a single "cell" from such a classification structure. Most Application Protocol developers see the use of the Application Protocol spread across many cells; the result is that even with a (supposedly) well-defined scope, there is constant scope creep.

### 6.3.4. Use of Application Contexts by Application Protocols

Application contexts, as embodied by the Application Domain Model, are an operational element of the standard in that they provided the basis for requirements gathering and decision-making, for the interpretation of data.

#### 6.3.4.1. Information Requirements Gathering

During the Application Protocol development, the purpose of Application Domain Model is to delineate the scope[9] of the Application Protocol such that:

- information used within the Application Domain can be identified;
- a decision can be made as to whether a piece of information is in scope or out of scope;
- of the in-scope information, a subset *of interest* can be specified as the information requirements that must be met by the Application Protocol.

It is impossible to identify all the bits of information used within an Industrial Application Domain. A subset of the information, however, can be documented as a model. Such a model constitutes the information requirements of the Application Domain. Information requirements can be met by many different mechanisms which combine a representation format/structure with a medium of storage or transmission: spoken language, written language, pictures, video, and digital data. Within ISO 10303 this model is a data model that is called an Application Reference Model (ARM).

The selection of the information *of interest* depends on the purpose of the Application Protocol - what is the Application Protocol supposed to *do*? As of the time of publication of this document, there is no formal component of an Application Protocol which states this purpose. A variant of the Application Activity Model called a **Usage Scenario** is informally used for this purpose; a Usage Scenario describes the use of the Application Protocol to perform a specific industrial task. A Usage Scenario is much, much narrower in scope than an Application Activity Model and there may be many Usage Scenarios associated with a given Application Protocol. Besides providing a statement of purpose for the Application Protocol (and thus the Application Reference Model), it provides a mechanism for the validation and testing of the Application Protocol.

#### 6.3.4.2. Context for Interpretation

Once the information requirements have been documented as an Application Reference Model, the next step is the specification of the mechanisms to meet the information requirements. The process of creating this specification is called *Interpretation.*

The constructs within the Integrated Resources (IRs) constitute a vocabulary of generalised and semantically "fuzzy" terms. The vocabulary is not specific to any particular Application Domain, but

[9] In many ways, the specification of the scope of an Application Protocol is very inadequate; prose scoping statement and an abstracted activity model are only two of many possible scoping mechanism

may be seen as applicable within a very wide Application Domain characterised by the intersection of the terms “industrial automation”, “product data” and “communication (of information)”. It is the Integrated Resource constructs that are used to create the data that constitute the “words” in a communication act.

“Interpretation” as used in the ISO 10303 development process is not some obtuse usage of a common English word. “Interpret” as used by ISO 10303 adheres very closely to the Dictionary definition of the word: “to explain or tell the meaning of.” Using the Integrated Resource constructs as a vocabulary, Integrated Resource constructs are *interpreted* as an element of the Application Reference Model - Integrated Resource constructs are used to satisfy the information requirements documented in an Application Reference Model. Thus, the Application Context provides the contextual description or explanation necessary to understand (i.e., interpret) data that is exchanged using an Application Protocol. The Integrated Resource constructs that are thus used by an Application Protocol constitute an Application Interpreted Model (AIM).

### 6.3.5. Application Protocol Integration

As subsets of the “real world”, Industrial Application Domains may partially overlap, may completely overlap, or may not overlap at all, depending on the perspective and purpose of the person or group recognising the domain. As a result, the Application Contexts/Scopes of Application Protocols may also overlap.

An overlap of Application Context indicates that there may also be an overlap of information requirements (though this is not necessarily the case; different information can be selected from the overlap of domains). When Application Protocol developers have identified the same information requirements within overlapping contexts, the development of a holistically consistent ISO 10303 standard mandates that the same information requirements be met the same way, i.e., interpret the Integrated Resource constructs the same way. The requirement for an interpretation of Integrated Resource constructs that are shared between two or more Application Protocols leads to the notion of an Application Interpreted Construct (AIC).

#### 6.3.5.1. Application Interpreted Constructs

If the Application Domains that interest the users of ISO 10303 never overlap, there would be no need for Application Protocol Integration. Application Protocol Integration is the process of identifying, harmonising, and satisfying in the same way the same information requirements in two or more Application Protocols. An Application Interpreted Construct (AIC) is the mechanism used to ensure that the same information requirements are met the same way.

An Application Interpreted Construct may be described as an “Application Interpreted Model fragment”. It is an interpreted subset of the Integrated Resources that is intended to meet an information requirement specified in two or more Application Protocols.

The need for an Application Interpreted Construct is recognised during the Interpretation process. Since the “vision” of the Interpretation extends across all Application Protocols and, therefore, all information requirements, participants in the process are in a position to recognise information requirements common to two or more Application Protocols. When such a need is recognised, an Application Interpreted Construct is formulated which documents a subset of the Integrated Resources for interpretation that is to be interpreted exactly the same in the development of the Application Interpreted Model of each Application Protocol. Since the information requirements may be documented differently in each Application Protocol (depending on the Application Reference Model modelling language chosen), the Application Interpreted Construct consists only of a small (short form) EXPRESS schema; interpretation of the Application Interpreted Construct schema still takes place and is documented as part of Application Interpreted Model development.

Application Protocol Integration is strictly a matter of establishing common interpretations for common information requirements. This is a subtle fact that is often missed, leading to efforts to Integrate Application Protocols through the harmonisation of the information requirements model (i.e., the Application Reference Model). See 7.3.5.3 on Harmonisation of Application Reference Models.

### 6.3.5.2. "Union” of Application Interpreted Models

Since the Application Interpreted Model represents the actual specification of the data that is exchanged using ISO 10303, it is sometimes said that the “real conceptual schema” in ISO 10303 is the union of all the Application Interpreted Models. This is not necessarily a meaningful statement. As presented above, the meaning of the elements within the Application Interpreted Model is very dependent on the Application Context/Domain of the Application Protocol. The union of Application Interpreted Models would imply the union of the Application Contexts/Domains and the union of Information Requirements. This would have the following consequences:

1. Since the Integrated Resource constructs are generic, reusable constructs, they may be interpreted differently within different Application Contexts/Domains. Combining Contexts/Domains would force

   i. some mechanism be included in the construct to differentiate between different uses (i.e., different semantics) of the construct; or

   ii. different entities to be created for each different use; thus instead of one construct serving different purposes/uses, there would be a different, unique construct for each purpose/use (which differs only slightly from the others, if at all).

2. The constraints on information entailed in each Application Context may conflict with constraints on the same information in other Application Contexts. The union of information requirements would force the resolution of all the conflicting constraints. Resolving conflicts would require relaxation or removal of the constraint, which inhibits the effective usage of the standard by removing meaning.

   EXAMPLE - Clash detection between piping system element and structural building elements requires that all elements have a shape representation defined; logical connectivity of a piping system does not require a shape representation.

Both of these consequences lead to the same result: a return to a semantically “flat” data exchange schema, just like IGES. The power of Application Protocols lies in the contextual information provided to a data exchange act, making the data semantically more precise, conformance more testable, and the communication act more effective and reliable. A statement like “Union of Application Interpreted Models” fails to recognise the contribution of the Application Context to inter system communication (e.g. data exchange) and promotes “old-fashioned” views of data exchange.

In explanations about the relationship between Application Protocols and Integrated Resources, it is sometimes said that the “union of Application Interpreted Models” is the “conceptual schema” of ISO 10303, alluding the ANSI/SPARC three-schema architecture. Setting aside the argument that “union of Application Interpreted Models” is not meaningful, within the architecture of ISO 10303 there is nothing that corresponds to the ANSI/SPARC notion of the conceptual schema; there is no single conceptual model for ISO 10303. Such a model/schema is neither useful or required for the purpose of inter system communication of product data. The Integrated Resources or an unconstrained “union of Application Interpreted Models” (if they are not equivalent) may be a single model and may be conceptual, but it is not intended to play the same role as the ANSI/SPARC conceptual schema.

#### 6.3.5.3. Harmonisation of Application Reference Models

The need to integrate Application Protocols is widely recognised and the subtlety of the integration that takes place during the Interpretation process has lead to integration efforts aimed at the Application Reference Model - integration of the information requirements. This is called Harmonisation and is intended to ensure that common requirements of two Application Protocols are met the same way (i.e., interpreted the same) by making (a portion of) the information requirements *model* (i.e., the Application Reference Model) the same.

Harmonisation efforts are typically spearheaded by Application Protocol developers focusing on allied Application Domains and result in efforts such as Core Models and Harmonisation workshops. While the workshops focus on “making the Application Reference Models the same”, the Core Model idea is an data model of generic constructs for a very broad Application Domain that can be reused and specialised by discipline-specific models for Application Domains that are encompassed by the broader Application Domain.

There are pros and cons against Harmonisation activities.

<u>Harmonisation workshops:</u>

- promote inter-Application Protocol awareness and foster a better understanding of the information requirements addressed by each Application Protocol;
- force each Application Protocol to document the information requirements more clearly and precisely; and
- externalise the shared sense that “Application Protocols do the same thing and should work together”.

However:

- information requirements are documented differently and Application Protocols are developed for different purposes, which makes harmonisation difficult;
- they require a great deal of effort and expense when the payback is negligible; Interpretation is still performed for each Application Protocol and similar requirements will be interpreted the same way, thus the Application Interpreted Models will be virtually identical regardless of the amount of harmonisation done;

<u>Core Models:</u>

- illustrate the relationship between members of a family of Application Protocols;
- provide a starting point, common understanding, and approach, for the development of a Application Protocols within a family, thus inhibiting Application Protocol development teams from “doing their own thing”;
- provide a good illustration of the functionality of the Application Protocols within a family.

However:

- the underlying approach of Core Models seems to duplicate the Integrated Resources;
- Core Models do not fit within the architecture of the ISO 10303;

- may restrict or undesirably channel Application Protocol development into a form that conforms with other Application Protocols in the family.

Since Application Protocol Integration is fundamentally done at the Interpretation stage, Harmonisation efforts are largely unnecessary. However, they are valuable for two important reasons:

- They provide visibility to developers and users that action is undertaken to ensure that Application Protocols that have common information requirements meet them the same way and can “interoperate” (see below).

- They provide clearer exposition of the relationship between the functionality of two or more Application Protocols.

#### 6.3.5.4. Application Protocol “Interoperability”

The phrase “Co-operative Use of Application Protocols” more accurately describes what is popularly known as “Application Protocol Interoperability”. This subject is addressed at length elsewhere in this document. The relationship to Application Contexts, however, is clear. Co-operative Use of Application Protocols implies that the Application Contexts of each Application Protocol overlap and that they have common information requirements.

This explanation of Application Contexts makes it clear (or clearer) that the Application Protocol development process is inherently intended to promote Application Protocol Integration, “Interoperability”, and co-operative use. “Interoperability” is only an issue because of the novelty of the Application Protocol use of Application Context and the subtlety of Interpretation.

### 6.3.6. Application Protocol Development Methodology

There are two changes related to the Application Context of an Application Protocol that could improve the Application Protocol development process:

- Incorporate harmonisation activities and tools;

- Use frameworks or taxonomies for Application Protocol scoping and planning.

Each of these were discussed at length above. It remains to be explained why these are not part of the Application Protocol development process.

Harmonisation activities are not part of the process because Application Protocol Integration has taken place as part the Interpretation process and there has not been much of a need for overt harmonisation activities. However, with the number of Application Protocols under development and planned, the burden on the Interpretation Project might make Harmonisation activities (and other planning activities) more important. The danger is in Harmonisation activities becoming the end-all and be-all of Application Protocol development; Harmonisation results only in harmonised information requirements - the actual specification is only realised after interpretation.

The use of frameworks and taxonomies in the development of Application Protocols hinges on one fact: no amount of planning within a voluntary organisation will prompt or direct individuals or teams to develop “needed” Application Protocols. It is always self-interest that motivates and animates the development of an Application Protocol.

Thus, there was a conscious decision NOT to delineate or identify Application Domains for the planning and development of Application Protocols. Rather, market interest and funding will produce Application Protocols to service the Application Domains which most critically need data exchange capabilities. The onus on ISO 10303 is to understand the Application Context of each Application Protocol and identify functional overlaps where the information requirements are the same.

# 7. Document structure

ISO 10303 is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application protocols, application interpreted constructs, abstract test suites, implementation methods and conformance testing. Table 1 below summarises these parts classes.

SPL 6/6/95: should this section be closer to the start of the document.

Table 1: ISO 10303 parts classes

| **Parts class** | **Contents** | **Readership** |
|---|---|---|
| Parts 1-9: Overview | General information about the structure and intent of the ISO 10303 family of standards | Everyone |
| Parts 11-19: Description methods | Definition of the structured methods and languages used in the development and definition of the ISO 10303 standard. | ISO 10303 developers, data modellers, system developers/ integrators |
| Parts 21-29: Implementation methods | Definition of the formats and methods to be used in exchanging or sharing product data using ISO 10303. | system developers/ integrators |
| Parts 31-39: Conformance testing methodology and framework | Definition of the requirements for conformance testing of ISO 10303 implementations. | testing laboratories, system developers, ISO 10303 developers, end users |
| Parts 41-49 and Parts 101-199: Integrated resources | Definitions of general purpose, context independent information units required to support the exchange of product data. These models may either be relevant to all application areas (integrated generic resources) or to an identified group of application areas (integrated application resources). | ISO 10303 developers, data modellers |
| Parts 201…: Application protocols | Definition of the context for the use of product data and specification of the use of the standard in that context to satisfy an industrial need. | system developers/ integrators, end users |
| Parts 501…: Application Interpreted Constructs | Definition of common requirement of integrated resource constructs across two or more industry applications. | ISO 10303 developers |
| Parts 301…: Abstract test suites | Definition of the abstract test cases to be used in conformance testing of implementations of corresponding Application Protocols. | testing laboratories, system developers/ integrators, end users |

Figure 3 below shows the relationship between the elements of the ISO 10303 architecture described in clauses 4 and 6, and the documentation of ISO 10303 as a standard. The elements of the architecture that are specific to an industrial application form the basis for Application Protocols: parts of ISO 10303 (200 series) that standardise the data specification for defined industry application semantics. Although Abstract Test Suites are specified for each Application Protocol they are (for historical reasons) published separately as parts in the 1200 series.

The elements of the architecture that are shared between applications are standardised either as Integrated Resources (40 series and 100 series) or as Application Interpreted Constructs (500 series).

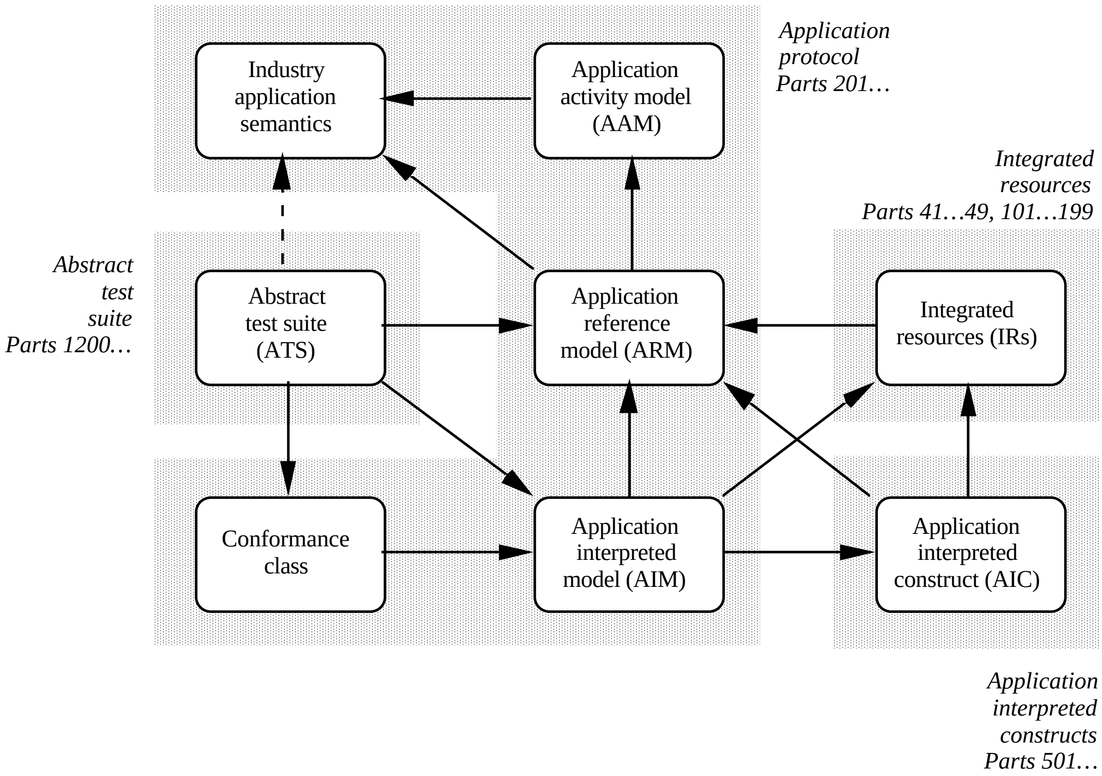


Figure 3: relationship of the ISO 10303 architecture to the documentation of the standard.

# SECTION II – Methodology

## 8. Integration and interpretation principles

Assigned to: WFD

### 8.1. Integration strategy

Text extracted from: Danner WG5 N106 Integration strategy

The resource integration method uses a strategy based on the use of the single product data architecture that is modular in construction. The principal goal of the resource integration strategy is to minimize the impact of change in the evolving standard while producing generalised, consistent, and canonical resource constructs for use in the development of application integrated models. The approach acknowledges the need to enhance the specification of resource constructs in response to new knowledge and extensions in scope. The resource integration method takes into consideration that the integrated resources are specified as a collection of documents. Each document has its own scope and unique content within the larger scope of ISO 10303 and other standards adhering to this methodology.

### 8.2. Introduction to interpretation

Text extracted from Gilbert/Yang Guidelines for AIM development (WG5 N108)

Interpretation is the process, mechanism, or manner by which meaning is assigned to an abstract representation of an event, object or concept. Within an Application Interpreted Model, the abstract representation is an EXPRESS construct. The meaning that is assigned to the construct may itself be defined by another construct. The interpretation of a construct defined in an integrated resource using another construct in an Application Interpreted Model may restrict, narrow, or constrain the semantic scope of the integrated resource construct, thereby *specializing* the construct.

The interpretation of a construct is ultimately grounded in human understanding. Interpretation draws not only on the meaning of the constructs themselves, but the contextual factors under which the constructs are generated, used, or received. Contextual factors in an Application Protocol may be defined by such things as the relationships of a particular construct to other constructs, or the overall application domain which specifies the scope of an Application Protocol. The integrated resources and the Application Reference Models are developed as representations of information in different (though established and related) contexts and are therefore subjected to the influence of different contextual factors. In order to account for contextual factors, interpretation relies on the human understanding of both the integrated resources and the Application Reference Model. The practice of interpretation of integrated resources in ISO 10303 (particularly the selection of integrated resource constructs) relies on human comprehension of the requirements represented in the Application Reference Model and the in-depth knowledge of the semantics and contextual factors under which the ISO 10303 resources are developed.

Comment from Stuart Lord 17/5/95 - some definition of “human understanding” required - logic?, knowledge?

Part of the interpretation process is ensuring the consistent interpretation for the same requirements found in different Application Protocols; this is accomplished through the identification and usage of application interpreted constructs (AIC). An Application Interpreted Construct is a module of conceptual constructs that is an interpretation of the ISO 10303 integrated resources and satisfies a specific set of application requirements. Inter-operability among various Application Protocols cannot be accomplished without consistent interpretation of the resource constructs.

## 8.3. The interpretation process

| |
|---|
| Text taken from Gilbert/Yang Guidelines for AIM development (WG5 N108) |

The application interpretation process is a formal and established part of the Application Protocol development process. ISO 10303 has defined standardised resources that are used as the basis for interpretation: the integrated resources and Application Interpreted Constructs. The application interpretation process includes activities such as: analyse Application Reference Model requirements, determine and select the corresponding resource constructs, develop the mapping table, and create the Application Interpreted Model EXPRESS annotated listings.

Within the ISO 10303 development organisation, the Application Interpreted Model development project and the individual Application Protocol projects are jointly responsible for the application interpretation activities; these activities are supported by the Application Protocol Integration project. A number of Application Interpreted Model development workshops are required to complete the interpretation for each application protocol project; the exact number of workshops depends on the complexity of the information requirements and the size of the scope for the application protocol's domain.

The Application Protocol Projects have the overall responsibility for the development of the Application Protocol. The Application Interpreted Model Development project is responsible for conducting the interpretation process and knowing the integrated resources. Together, an Application Protocol Project team and the Application Interpreted Model Development project team construct the Application Interpreted Model by interpreting the integrated resources; the Application Protocol project team brings to the process an understanding of the domain information requirements and the Application Interpreted Model Development team brings an understanding of the integrated resources and the interpretation process. The Application Protocol Integration project supports the interpretation process by supplying Application Interpreted Constructs.

# 9. Application protocol development

| |
|---|
| Text taken from AP Guidelines section 4.<br><br>Editorial instruction: this clause will contain description of the Application Interpreted Construct development method. |

## 9.1. Methods overview

| |
|---|
| Assigned to: JPF |

### 9.1.1. Definition of Scope and Requirements

The ISO 10303 methodology is closely focused on the development of Application Protocols (APs), i.e., standardised data specifications that satisfy identified industry needs. The development process for an Application Protocol is initiated by the identification of such a need; this may arise from collaborative projects in industry, trade associations, standards bodies, individual companies, etc. Some Application Protocol proposals arise from Application Protocol Planning Projects (APPPs) within the ISO ISO 10303 committees; these projects generally focus on the needs for multiple Application Protocols (or very large Application Protocols) within an industry sector.

The first stage in articulating this need is the definition, at a high level, of the scope of the proposed Application Protocol and the requirements that it is intended to fulfil. This definition not only enables validation of the proposed Application Protocol by potential users and implementors in industry, but also its assessment for overlaps and redundancies with other Application Protocols.

The basis for the definition of the scope and requirements for a proposed Application Protocol is the identification of the Industry Application Semantics that are to be standardised. This is characterised by the types of products to be described, the kind of product data used, the disciplines that make use of the product data, the life-cycle stages in which the data is created and used, etc. This characterisation is extended through the definition of a Application Activity Model that elaborates the activities and information flows that are to be in the scope of the Application Protocol. Although other methods for activity modelling are permitted, the IDEF0 methodology is almost universally used in ISO 10303. The formal specification of this activity model is often accompanied by the formulation of usage scenarios: informal (but detailed) descriptions of the intended use of the Application Protocol.

At this stage in the development of an Application Protocol, a high level data model (or “data planning model") may be produced as an aid to understanding and analysis of the scope of data to be supported. Such a model attempts to capture the subject areas, or major groups of data, that are in scope. No specific method for the development of such models is mandated; most are documented using simple, informal graphical presentations.

On the basis of the initial statement of scope and requirements, the proposed Application Protocol is balloted as a New Work Item Proposal under ISO rules; if approved, the development of the Application Protocol as a part of ISO 10303 is mandated.

### 9.1.2. Information requirements

The second phase in the development of an Application Protocol is the discovery and documentation of the detailed information requirements that are to be fulfilled. It is important to note that these requirements are discovered, rather than defined: the requirements already exist as the data that underlies industry practices, processes, and systems. These requirements are analysed and documented through the development of an Application Reference Model (Application Reference Model). The term “Application Reference Model” may be used to refer to two elements of the Application Protocol:

- English language statements of the information requirements, in the form of defined application objects, attributes, and relationships; these application objects may be grouped into Units of Functionality (UOFs);

- a graphical presentation of these requirements, using notations such as IDEF1X, NIAM, or EXPRESS-G.

JPF 3/95: The development of an ARM, and the specification of information requirements, is one of the weaker elements of the ISO 10303 methodology, in that the guidance provided to Application Protocol development teams is little more than that presented above. Significant advances are, however, being made in this area, particularly within projects that are addressing a broad spectrum of requirements within an industry sector. Improved techniques for ARM development are being employed in Application Protocol projects in the automotive, process plant, shipbuilding, and building & construction sectors; harmonisation and acceptance of these techniques is likely to lead to their incorporation into the “core” methodology of ISO 10303.

Once the Application Reference Model is complete to the satisfaction of the Application Protocol development team, full documentation of the scope (Application Activity Model) and requirements (Application Reference Model) is distributed as a Committee Draft for Comment (CDC); this process is designed to ensure adequate and effective review and validation of the Application Protocol by experts in industry.

### 9.1.3. Application interpretation

The first two phases in the specification of an Application Protocol are undertaken by the project team responsible for its development. From this point onwards, however, the further development of the Application Protocol is undertaken through synergy between the project team and the “core” functions of ISO 10303: Application Interpreted Model Development, Application Protocol Integration, Resource Integration, and ATS Development. This interaction may be seen as part of a “matrix” management approach to the development of the standard: each Application Protocol results from the definition of requirements by industry or application experts, the fulfilment of those requirements by “ISO 10303” experts, and the validation of solutions by the industry experts.

The first of these synergistic phases is the development of the Application Interpreted Model: the creation of a data specification based on the Integrated Resources that meets the requirements stated in the Application Reference Model. This phase begins with analysis of the Application Reference Model by the Application Interpreted Model Development team: this analysis focuses on gaining deep understanding of the application requirements, and relating this understanding to the underlying concepts of the Integrated Resources.

A second part of this analysis, undertaken by the Application Protocol Integration team, is the identification of overlaps with other, existing Application Protocols. Where such overlaps correspond to shared requirements across two or more Application Protocols, the development and use of Application Interpreted Constructs, i.e., a shared solution to the common requirements, is enabled.

The third aspect of this analysis is the identification of requirements that are not supported by the Integrated Resources, and therefore give rise to a need for extension to the Integrated Resources. It is an important principle of the ISO 10303 methodology that Application Protocols do not themselves define extensions to the resource models. These extensions are developed according to the Resource Integration method.

### 9.1.4. Mapping

Following these analyses, the process of application interpretation involves the identification of the mapping from each information requirement (application object, attribute, or relationship) to one or more constructs from the Integrated Resources. This mapping results in the creation of two elements of the documentation of an Application Protocol:

- a Mapping Table, that specifies the precise mapping of each application requirement;
- the Application Interpreted Model (AIM).

As each application requirement is mapped, the result of the mapping is incorporated into the Application Interpreted Model: the interpretation process results in the creation of a new data model (EXPRESS schema) from the Integrated Resources. Within this new data model, each construct acquires the context of the Application Protocol; in many cases, the requirements of this context are fulfilled by applying constraints to the constructs mapped from the Integrated Resources. However, even when a constraint is not explicitly specified, this additional contextual information means that an entity definition in an Application Interpreted Model is not the same as an apparently identical definition in the Integrated Resources. The definition in the Application Interpreted Model represents a usage of the resource construct, and refines its meaning for the context of the Application Protocol.

### 9.1.5. Application Interpreted Models

An Application Interpreted Model is specified as a “short form” EXPRESS schema: this consists of the EXPRESS interface statements that select constructs from the Integrated Resource schemas, together with the additional specialisations and constraints that are defined by the mapping process. A second specification (the “annotated listing") provides refinements to the natural language definitions of the constructs selected from the Integrated Resources.

The schema interfacing capabilities of EXPRESS mean that, through the use of suitable software tools, all references in the Application Interpreted Model short form can be resolved to create a single schema form of the Application Interpreted Model, known as the “long form”. Algorithmically: Short form + Integrated Resources --> Long form

This long form Application Interpreted Model is provided in electronic form as part of the Application Protocol documentation, and is the basis for implementations based on the ISO 10303-21 file format.

### 9.1.6. Application Interpreted Constructs

The potential for creation and use of Application Interpreted Constructs (AICs) is described above as part of the initial phase of the application interpretation process. Application Interpreted Constructs arise where different applications share functional requirements, which may in some cases relate to the use of common or similar computer systems; many of the Application Interpreted Constructs identified to date relate to common uses of geometric representation (B-Rep, Surface models, etc.), across different application areas. For example, if automotive design, sheet metal tooling, and shipbuilding all use surface models to represent product shape, then a common Application Interpreted Construct for this functional requirement can be developed and used.

The existence of an Application Interpreted Construct identifies the potential for reuse of implementation code, and for the sharing of data between applications. This latter point is particularly important: since the data specification for surface models is common across the three applications identified above, data instances may be shared between them.

> NOTE – However, just because they may be shared does not necessarily mean that it is useful todo so. In this example, it is easy to see that the same surface model might be used by the automotive design and sheet metal tooling applications; similar sharing between automotive design and shipbuilding is less likely to be useful.

The method for the development of Application Interpreted Constructs is broadly similar to that for Application Interpreted Models, i.e., the selection and constraint of Integrated Resource constructs. The only significant difference is that each Application Interpreted Construct includes one or more “root” entity, that carries the constraints relevant to the Application Interpreted Construct; a root entity acts as a scoping mechanism for the applicability of these constraints when the Application Interpreted Construct is used within an Application Interpreted Model. The method for use of Application Interpreted Constructs is simple: an Application Interpreted Construct is used through inclusion, without modification or constraint, within an Application Interpreted Model. It is this lack of modification or constraints in the use of an Application Interpreted Construct that ensures compatibility across the Application Interpreted Models that use an Application Interpreted Construct.

### 9.1.7. Resource Integration

Analysis of the requirements specified in an Application Protocol may identify the need for extension to the ISO 10303 Integrated Resources. This approach to Integrated Resources represents the “mature” phase of ISO 10303 development; previously, complete, existing models proposed as ISO 10303 resources have been “integrated” with the core Generic Product Data Model; it is this “creation” phase that has given rise to the Integrated Resources as they are today. As with Application Interpreted Model development, resource integration is a synergistic process, involving the interaction between the experts in the discipline covered by the resource model (geometry, finite element analysis, etc.), and the ISO 10303 integration team.

The creation phase requires a more involved and complex method, since the requirements for extension to the Integrated Resources have been large, and often expressed as complex, mature EXPRESS data models. However, where these models have been developed without the use of the basic ISO 10303 architecture, considerable restructuring of these “Draft Resource Models” has proved necessary.

The resource integration method involves several phases; these may be summarised as follows:

- **analysis:** comparison of the requirements underlying the draft resource model (or those identified within an Application Reference Model) with the concepts of the ISO 10303 Integrated Resources; where a draft resource model exists, this analysis also includes a comparison of specific data model structures;

- **restructuring:** where a draft resource model is the source of the requirements for extensions, this model is restructured so that it fits semantically and structurally with the existing Integrated Resources, and conforms to the EXPRESS usage practices adopted within the integration process;

- **verification:** the model that results from the integration process is verified by appropriate application and discipline experts to ensure that requirements are accurately and completely fulfilled.

Where the requirements for extension to the Integrated Resources arise from the application interpretation process, the restructuring phase is trivial, since the extension is designed and created by the integration team itself (i.e., in this case “restructuring” could be replaced by “creation").

### 9.1.8. Requirements for implementation and testing

The various methods described in sections 6.1 to 6.4 above relate to the creation of data specifications within ISO 10303. It must not be forgotten, however, that these specifications are useful only as the basis for implementation of data exchange or sharing, and that such implementations are required to be testable. Requirements for implementation and testing are fulfilled through the specification of conformance classes within an Application Protocol, and of an Abstract Test Suite for each Application Protocol.

#### 9.1.8.1. Conformance classes

Conformance classes are developed through the analysis of the usage scenarios identified in the initial phase of the development of the Application Protocol, and through an understanding of the capabilities of the computer applications that are expected to support the Application Protocol. Each conformance class defines a fixed boundary for the scope of an implementation; this is determined on the basis of defining a subset of total capability of the Application Protocol that is practical to implement whilst not comprising the industry application semantics that define the purpose for the existence of the Application Protocol.

#### 9.1.8.2. Abstract Test Suites

Abstract Test Suites are developed through analysis of the requirements specified in the Application Reference Model and the Application Interpreted Model of an Application Protocol. An ATS has several constituent components:

- **test purposes:** formal statements of the aspects of an Application Protocol; these are derived directly from the requirements specified in the Application Protocol, and are the basis for verdict assignment during conformance testing;

- **verdict criteria:** the basis for determination of success, failure, or uncertainty with respect to the results of testing;
- **abstract test cases:** parameterised forms of simple (yet representative) test cases that exercise one or more test purposes.

The detailed methods for the development of Abstract Test Suites have reached stability over the last six months: initial Abstract Test Suites are to be published for initial review during 1995.

### 9.1.9. Approval and publication

The completion of an Application Protocol, Integrated Resource, or Abstract Test Suite initiates the formal processes of review and approval as ISO Committee Drafts (CD) and Draft International Standards (DIS). Response to comments raised during these reviews gives rise to iterative application of the methods outlined above.

## 9.2. Application protocol development procedures

JPF 6/6/95: these procedural tables are becoming difficult to manage. We should discuss at Washington whether the purpose for which the tables were created would be better served by a more formal approach, combining IDEF0 formalisms (mostly present in existing WG5 documentation) plus ISO 9000 style procedure descriptions.

SPL 8/8/95: In the absence of IDEFO diagrams I have taken the procedural text from the tables and put them as an itemised list with a reference in the table to the appropriate item. This should make them easier to manage in the short term.

The table below summarises the process of development ISO 10303 Application Protocols, including the development of Abstract Test Suites, Application Interpreted Constructs, and Integrated Resources. It is assumed for the purpose of this summary that Application Interpreted Constructs and Integrated Resources are developed only as required by Application Protocols.

The “type” column classifies each procedure according to the type of procedure, as follows:

- project management and planning (P): of the Application Protocol project;
- approval (A): internal SC4 procedures used to approve the development of an Application Protocol in the context of other parts of ISO 10303;
- development (D): technical development tasks undertaken by the Application Protocol project team;
- training (T): provided to the Application Protocol project team by other functions within SC4;
- integration (I): shared technical development between the Application Protocol team and Application Interpreted Model development and/or Application Protocol integration functions;
- qualification (Q): quality checks;
- documentation (E): development of the Application Protocol document, and accompanying reports.

| | Type | Method(s) | Architectural element(s) | Documentation element(s) |
|---|---|---|---|---|
| 1 | P, A | N/A<br>*NOTE: the adoption of the ISO NWI proposal procedure by SC4 changes this phase of Application Protocol development.* | | |
| 2 | A | N/A<br>*What is the role of the Application Protocol coordinator with respect to use of the NWI proposal procedure?* | | |
| 3 | A | N/A<br>*Comments regarding procedure changes above apply.* | | |
| 4 | D | activity/process analysis and modelling<br>IDEFO, other methods?<br>Data dictionary for activities and flows | data architecture<br>integration architecture<br>Application Protocol integration architecture<br>Application Protocol framework(s) | |
| 5 | P, D, E | as above | as above | |
| 6 | A | N/A<br>*Effect of NWI proposal procedure?* | N/A | |
| 7 | A | N/A<br>*Effect of NWI proposal procedure?* | N/A | |
| 8 | T | Training in technical writing and reviews, ISO and SC4 guidelines | | Document architecture |
| 9 | V | Review | N/A | |
| 10 | D | Data discovery<br>Data modelling<br>IDEF1X, NIAM, EXPRESS(-G), other E-R modelling methods<br>Re-use of UOFs[10]<br>“Harmonisation” with Application Protocols with overlapping scopes | Application Protocol Frameworks?<br>Reference models?<br>Core models? | |
| 11 | I | Application Interpreted Model development<br>Application Reference Model analysis<br>Identification of resource models and Application Interpreted Constructs to be used<br>**Link to Ressource modelling and resource integration procedures**<br>Application Protocol Integration<br>Application Interpreted Construct Development<br>Application Interpreted Construct Usage | Data architecture<br>Integration architecture<br>Application Interpreted Construct architecture<br>*Note: care needs to be taken in using “architecture” to refer to both structures and populated structures!* | |
| 12 | D (,V) | Usage test development<br>Test purpose derivation | N/A | |
| 13 | Q | Part Qualification<br>Part editing | | Document architecture<br>Qualification manuals<br>Supplementary directives |
| 14 | P, E, V | N/A | N/A | |
| 15 | P | N/A | N/A | |

[10] APIP keep AIC library - who keeps UoF library? (Stuart Lord comment 17/5/95)

| | | | | |
|---|---|---|---|---|
| 16 | I | Application Interpreted Model development<br>Selection and adaptation of Integrated Resource constructs<br>Identification and use of Application Interpreted Constructs | Integration architecture<br>Application Interpreted Construct architecture | |
| 17 | I | Mapping table development and documentation<br>EXPRESS usage (Application Interpreted Model development guidelines)<br>Syntactic and semantic rules | Integration architecture<br>Application Interpreted Construct architecture | |
| 18 | D | EXPRESS usage<br>Test purpose development | Data architecture<br>Implementation architecture (short form vs. long form) | |
| 19 | D | Conformance testing methodology | Implementation architecture<br>Conformance testing framework? | |
| 20 | E, Q | N/A | N/A | |
| 21 | E, A | N/A | N/A | |
| 22 | A | N/A | N/A | |
| 23 | A | N/A | N/A | |

1. Industry representatives document requirements for Application Protocols and develop a proposal for a ISO 10303 Application Protocol planning project (see clause A.1). The proposal shall be submitted by a SC4 member country to the Application Protocol Coordinator for the PMAG. The Application Protocol planning project proposal describes the scope of the project, i.e., type(s) of product, application tasks, discipline views, types of product data, and stages of the product life cycle to be included, evidence of international industry need for Application Protocols in this area, committed human resources to the project, overlaps and relationships to other Application Protocols, and the schedule for delivering Application Protocol project proposals.

2. The Application Protocol Coordinator reviews the proposal to ensure that the proposal is complete and distributes each complete proposal to the PMAG for review and approval. If the proposal is incomplete or ambiguous, the Application Protocol Coordinator returns the proposal to the submitter with recommendations for improvement.

3. The PMAG reviews the proposal to ensure that the proposed Application Protocol planning project: 1) is compatible with existing Application Protocol planning projects and Application Protocol projects, 2) will address industry requirements shared by many organizations and countries, and 3) has sufficient human resources to complete the planned work. Based on the results of this analysis and the recommendations of the WG conveners, the PMAG decides whether to approve the proposed Application Protocol planning project.

4. Industry representatives, application experts, and vendors of computer-aided tools analyze and document requirements and priorities for Application Protocols within a specific application domain. With the assistance of ISO 10303 experts, application experts assess the correspondence of the application requirements to: 1) existing Application Protocol planning project and Application Protocol projects, and 2) the scope and architecture of ISO 10303. These analyses provides the basis for an initial definition of the scope of an Application Protocol or suite of Application Protocols and a development plan.

5. Industry representatives and application experts complete an Application Activity Model, a representative set of usage scenarios, and an Application Protocol project proposal (see clause A.3). The Application Protocol project proposal describes the scope of the project, evidence of international industry need for the Application Protocol, committed human resources to the project, overlaps and relationships to other Application Protocols, and the schedule for completing the Application Protocol. It is recommended that Application Protocol project proposals be defined as part of an Application Protocol planning project. A member body of SC4 shall submit the proposal to the Application Protocol Coordinator.

6. The Application Protocol Coordinator reviews the proposal to ensure that the proposal is complete and distributes each complete proposal to the PMAG for review and approval. If the proposal is incomplete or ambiguous, the Application Protocol Coordinator returns the proposal to the submitter with recommendations for improvement.

7. The PMAG reviews the proposal to ensure that the proposed Application Protocol project: 1) is compatible with existing Application Protocol projects, 2) will address industry requirements shared by many organizations and countries, and 3) has sufficient human resources to complete the planned work. The PMAG uses this analysis and the recommendations of the WG conveners and the PMAG members to determine whether to approve the proposal as a SC4 Application Protocol project. After the PMAG approves the Application Protocol project, SC4 votes to approve the project.

8. The Application Protocol project team meets with the Part Qualification and Validation Project and the Editing Committee for instruction on how to efficiently and correctly develop and document Application Protocols.

9. The Application Protocol project team conducts industry reviews and evaluations of the Application Protocol scope, Application Activity Model, and requirements. The results of these reviews and the basis for industry acceptance are documented in a scope and requirements evaluation report. This report is included as a clause of the Application Protocol validation report. Issues defined during these reviews shall be documented in the Application Protocol issues log.

10. The Application Protocol project team uses the Application Activity Model, scope, and requirements as the basis for defining the application reference model (Application Reference Model) and the units of functionality (UoFs).

11. The Application Protocol project team submits the information requirements, Application Reference Model, and UoFs for review and comments by the WG4 Application Interpreted Model Development Project (ADP) and the WG4 Application Protocol Integration Project (APIP). The APD assists the Application Protocol project in the development of the Application Interpreted Model and determines whether additions to the existing integrated resources are required to meet the information requirements. If additions to the integrated resources are required, the APD and the Application Protocol project team define a plan for developing these additions.The APIP is responsible for the identification of Application Interpreted Constructs and management of the Application Interpreted Construct library. The Application Protocol project teams receive guidance on the use and documentation of Application Interpreted Constructs from APIP.

12. The Application Protocol project team validates the Application Reference Model and UoFs and produces the Application Reference Model validation report. This report is included as a clause of the Application Protocol validation report. The Application Protocol project team produces usage tests as part of validating the Application Reference Model.

13. The Application Protocol project team submits Group 1 of the Application Protocol to the Qualification and Validation Project for initial qualification review and to the Editing Committee for initial editorial review.

14. The Application Protocol project team documents the plan for resolving comments and issues raised in the WG4 reviews and submits the completed Group 1 of the Application Protocol to the SC4 chair for distribution to the SC4 members as a Committee Draft for Comment (CDC).

15. The Application Protocol project team reviews the comments from the SC4 CDC and develops a plan for resolving issues raised.

16. The ADP, working with the Application Protocol project team, selects and interprets resource constructs for the Application Interpreted Model.

17. The Application Protocol project team, with ADP assistance, produces the mapping table, the Application Interpreted Model (EXPRESS and EXPRESS-G), and the integrated resources interpretation report. The interpretation report is included as a clause of the Application Protocol validation report.

18. Application Protocol project team compiles and validates the Application Interpreted Model and produces the Application Interpreted Model validation report. This report is included as a clause of the Application Protocol validation report.

19. The Application Protocol project team defines conformance requirements and implementation method specific requirements.

20. The Application Protocol project team completes the Application Protocol document and submits the document, i.e., both Group 1 and Group 2, to the ADP for approval and to the QVP for final qualification approval.

21. The Application Protocol project team resolves all qualification issues, approves the Application Protocol to Project Draft status, and submits the Application Protocol to the convener of the parent WG and to the WG4 Convener for approval. The Application Protocol issues log and the Application Protocol validation report are circulated with the Application Protocol for these reviews.

22. The Application Protocol project team submits the Application Protocol to the Editing Committee for review and approval.

23. The Application Protocol project team submits the Application Protocol to the PMAG for review and approval as an ISO committee draft (CD).

NOTE: This is not the end of the complete process. Two major elements are missing: the completion of the CD and DIS ballot processes (including rework of material covered in tasks 9-23), and development of the Abstract Test Suite. The recently confirmed SC4 requirement for development of prototype implementation(s) needs to be taken into account as well.

## 9.3. Industry Application Semantics

Assigned to: WFD

## 9.4. Application Activity Model development methods

Assigned to: WFD

This text taken from APG 4.1 Development and review of the scope and information requirements.

The first phase of developing an Application Protocol is the definition of its scope and information requirements. Definition of the scope and information requirements begins with the formulation of a statement of the application context and functional requirements for the Application Protocol. This statement shall define the type(s) of product, the stages in the life cycle of the product(s), the product data application(s), and the use of the product data within the application(s) targeted for the Application Protocol. The detailed scoping and information requirements definition shall proceed from this statement.

Scope definition shall be refined via the development of an application activity model (AAM). The Application Activity Model describes the input and output information requirements of the processes within the application context. The Application Activity Model shall be documented with the process modelling technique IDEF0 [5]. The Application Activity Model shall include a glossary that defines all activities and elements in the model.

The identification of product data usage scenarios pertinent to the application and example parts or products that will be represented with the Application Protocol shall be used as aids to defining detailed data requirements. As the Application Activity Model and information requirements become more detailed, the scoping statement prepared at the beginning of the scoping phase shall be updated to correspond.

The Application Protocol's scope, Application Activity Model, and information requirements shall be carefully defined and documented. This documentation, in addition to the example products and product data usage scenarios, provides the foundation for developing the Application Protocol. The usage scenarios are extremely valuable in the subsequent validation of the Application Reference Model and the Application Interpreted Model.

At each level of decomposition of the Application Activity Model, the activities, inputs, controls, outputs, and mechanisms should be examined, and a determination made as to whether they are in scope for the Application Protocol. The inclusion of mechanisms, e.g., resources and tools, in an Application Activity Model should be carefully analyzed. Often, mechanisms are organization or enterprise dependent, and those dependencies shall not be included in the Application Activity Model. Only those mechanisms that are organization and enterprise independent shall be included. The Application Protocol development process is designed to develop organization and enterprise independent models.

An Application Activity Model for the existing “as is” processes should be developed before defining an Application Activity Model for future “to be” processes. The decision to scope an Application Protocol based on future processes rather than existing processes requires careful analysis of the industry need(s), objectives sought by the proposed improved process, the time required to deliver the Application Protocol to industry, and the relative costs and benefits.

The Application Protocol scope statement shall include a summary of the type(s) of product, the application processes, the types of product data, and the discipline views of the product that are within the scope. For clarification, the scope statement may also identify the type(s) of product, the application processes, the types of product data, and the discipline views of the product that are outside of the scope.

As the scope, requirements, and Application Activity Model are further defined, the Application Protocol team should conduct fitness testing and evaluations of these items by experts in the application. The majority of these expert reviewers should not have participated in the Application Protocol development. Representatives from all relevant industries and from a broad spectrum of user organizations should be included to the maximum extent possible.

The objectives of this review are to ensure that:

1. the scope and requirements are accurate, viable, and meet a significant industrial need;

2. the Application Activity Model accurately represents all areas which are described by the scope statement, and

3. the documentation is sufficient, unambiguous, and conveys the correct meanings.

All issues raised during the review(s) and the resolutions of these issues shall be documented in the Application Protocol's Scope and Requirements Evaluation Report. This report shall include the list of workshops, surveys, and reviews used in the assessment, the identification of the organizations participating in the reviews, the list of the expert reviewers, with their qualifications, summaries of the review(s) and the evaluations, and an issues log with the issues resolved. The report shall be included in the Application Protocol validation report.

## 9.5. Application Reference Model development methods

Assigned to: YY/WCB

This text taken from: APG 4.3 Development and review of the application reference model

Comment from Stuart Lord - Section not particularly helpful. Does reasonable job of stating what has to be done, but doesn't give much guidance on how to do it or how to recognize when the job is done, not to mention the issue of whether the people are competent to do it.

When the detailed scope and functional requirements have been defined, the information domain of the Application Protocol shall be defined by the use of the application reference model (ARM). The Application Reference Model shall be developed using a formal data description language, i.e., EXPRESS, IDEF1X, or NIAM. Application Reference Models specified using EXPRESS shall include an EXPRESS-G presentation. Each application information requirement that is within scope in the Application Activity Model shall be expressed in the Application Reference Model. Conversely, each element of the Application Reference Model shall satisfy a documented information need of the application. The Application Reference Model shall describe fully the data needs of the application, using the terminology of the application.

An Application Reference Model shall be sufficiently detailed so that the selection and interpretation of the integrated resources can be done accurately. The Application Reference Model documents the required data and relationships. The graphical presentation of the Application Reference Model, i.e., EXPRESS-G, IDEF1X, or NIAM, aids the understanding and review of the information requirements and definitions. The Application Reference Model diagrams shall be at a detail level sufficient to present the requirements in a manner that it is understandable to an application domain expert. The information requirements shall be modelled only to the level necessary to convey the information that is important from the application experts' point of view.

Stuart Lord comment. Does this mean “find an application expert who agrees with the data model” or “get the application expert to do his own data model"!

A mechanism for modularizing the scope of an Application Protocol into manageable constructs is to define Units of Functionality (UoF). A UoF is a collection of application objects and assertions that conveys one or more well-defined concepts within the context of an Application Reference Model. A UoF usually supports an application function or process. UoFs are used to organize and summarize the functionality of the Application Reference Model. For example, if a geometric modelling application has a requirement for wireframe geometry, then a UoF may be defined which provides a grouping of those application objects in the Application Reference Model which are intended to support geometric modelling using wireframe geometry. UoFs are models of aggregates of data which are important to the application domain experts.

As the Application Reference Model is developed, the constructs which correspond to each UoF shall be grouped together so that they are readily identifiable. A list of the UoFs with definitions shall be maintained. This list shall include the application processes of the Application Activity Model that require each UoF. Documenting UoFs facilitates the integration of Application Protocols at the Application Reference Model level.

As the Application Reference Model is refined, traceability and consistency shall be maintained between the scope, functional requirements, Application Activity Model, and Application Reference Model. The Application Protocol development team shall ensure that each requirement identified in the Application Activity Model is expressed in the Application Reference Model.

Application Protocol projects shall maintain knowledge of the domain and status of other Application Protocol projects[11] and shall investigate possible overlaps with other Application Protocols and

[11] see STEP Application Protocols Status and Summary Report, ISO TC184/SC4 PMAG, by Halvorson and Palmer.

Application Protocol projects. This shall include the analysis for common information requirements, similar units of functionality, and correspondence between Application Reference Models. The Application Interpreted Model Development Project and Application Protocol Integration Project also conduct such analyses to identify overlaps among Application Protocols. Overlaps are evaluated for Application Protocol integration requirements.

Application Protocols with similar UoFs or information requirements shall be compared semantically by the Application Protocol Integration Project to determine functional equivalence. When two or more Application Protocols have equivalent UoFs or common information requirements, the same interpretation of the integrated resources shall be used in the Application Protocols' Application Interpreted Models. This interpretation shall be accomplished by the inclusion of a common module in each of the Application Protocols. This common module is called an application interpreted construct (AIC)[12]. Application Interpreted Constructs and a list of the Application Interpreted Models that use each Application Interpreted Construct shall be included in the Application Interpreted Construct Library. The Application Interpreted Construct Library is maintained by the Application Protocol Integration Project. This method will provide for consistency of ISO 10303 data representation among Application Protocols.

After the Application Reference Model is developed it shall undergo validation to ensure that it fulfils the functional requirements, is self-consistent, and covers the scope of the application completely. This validation should be done by both expert modellers for integrity testing and application experts for fitness testing. The functionality documented in the Application Reference Model shall meet the requirements expressed in the scope. Additionally, the functionality documented in the Application Reference Model shall not exceed the boundaries defined in the scope and the requirements.

The example parts and the input and output data requirements used in the initial scope and requirements definition should be used for building the usage tests for validating the Application Reference Model. Usage tests document typical operations for creating or accessing product data. The set of Application Reference Model usage tests should be carefully defined to ensure coverage of the application context and functional requirements, the information defined in the Application Reference Model, and possible combinations of product representations. Difficulty in defining a meaningful usage test which exercises an application object may indicate that the application object is not needed.

Once the Application Reference Model is complete and has been validated with the usage tests, the Application Protocol team shall submit the Application Reference Model and the Application Reference Model validation report for qualification. Personnel shall be assigned to work with the Application Protocol project to ensure that the model(s) is correct and complete[13]. Upon completion of the review, the Application Protocol project team shall determine how to resolve the comments and the schedule for distributing the Group 1 of the Application Protocol for review and comment by SC4. The review comments, the Application Protocol project's plan for resolving all issues, and the completed clauses of the Application Protocol validation report shall be included with the Group 1 distribution. The SC4 review of the Group 1 Committee Draft for Comment (CDC) provides a mechanism to ensure international consensus on the scope and requirements before resources are expended in developing the Application Interpreted Model and other related documentation.

### 9.5.1. EXPRESS Application Reference Models

| Text taken from: APG 4.3.1.1 EXPRESS ARMs |
|---|

If the Application Reference Model has been modelled using EXPRESS, then the application object definitions and application assertions shall be specified as follows:

---

[12] see AP Integration Practices: AIC Development, ISO TC184/SC4/WG4 Document N53, October 1992

[13] See STEP application protocol qualification manual, ISO TC184/SC4/WG4(P5) Document N502, November 1993.

- Each entity shall be stated in the application object definitions.

- Each attribute whose data type is either a base data type or a defined data type which is not a SELECT data type with a select list that contains entity types or other select types with select lists that contain entity types shall be stated as an ATTRIBUTE of that entity in the application object definition.

- Each attribute whose data type is an aggregate of either a base type or a defined type which is not a SELECT data type with a select list that contains entity types or other select types with select lists that contain entity types shall be defined as an ATTRIBUTE in the application object definition, with the cardinality defined in the definition.

- Each attribute whose data type is an aggregate of either an entity type or a SELECT type with a select list that contains either entity types or other select types with select lists that contain entity types shall be stated in the application assertions with the cardinality defined by the aggregate bounds.

- Each attribute whose data type is an entity type shall be stated as a RELATIONSHIP between the two entities in the application assertions.

- Each attribute whose data type is a SELECT data type with a select list that contains entity types or other select types with select lists that contain entity types shall be stated as a RELATIONSHIP between the entity containing the attribute and each of the options in the SELECT list in the application assertions.

- The INVERSE statements of referenced entities shall be examined in order to specify the cardinality constraints in the application assertions.

### 9.5.2. IDEF1X Application Reference Models

Text taken from: APG 4.3.1.2 IDEF1X ARMs

If the Application Reference Model has been modelled in IDEF1X, then the application object definitions and application assertions shall be specified as follows:

- Each entity shall be stated in the application object definitions.

- Each attribute shall be stated as an ATTRIBUTE in the application object definitions.

- The business rule defining each relationship shall be stated in the application assertions.

- The notes which document any additional constraints shall be stated in the application assertions.

### 9.5.3. NIAM Application Reference Models

Text taken from: APG 4.3.1.3 NIAM ARMs

If the Application Reference Model has been modelled in NIAM, then the application object definitions and application assertions shall be specified as follows:

- Each NOLOT shall be stated in the application object definitions.

- For each NOLOT, each LOT connected directly by a bridge (FACT TYPE between NOLOT and LOT) shall be stated as an ATTRIBUTE of the entity which was stated for that NOLOT in the application object definitions.

- If the role adjacent to the LOT has no ALWAYS or MANDATORY constraint, then the LOT or ATTRIBUTE is OPTIONAL and shall be stated as such in the attribute definition.

- If the role adjacent to the LOT has a SIMPLE UNIQUE constraint, then the LOT or ATTRIBUTE is UNIQUE and shall be stated as such in the attribute definition.

- If two or more LOTs share a JOINT UNIQUE constraint, then the LOTs or ATTRIBUTES are jointly UNIQUE and shall be stated as such in the definitions for each attribute.

- Each IDEA shall be stated as a RELATIONSHIP between the NOLOT corresponding objects in the application assertions.

### 9.5.4. Application Reference Model validation

| Text taken from APG 4.3.2 ARM validation |
|---|

The Application Protocol team shall summarize its plan for Application Reference Model validation and the validation results in an Application Reference Model validation report. This report shall include the rationale for the selection of representative test parts and usage tests and an analysis of the degree of coverage provided by the validation testing. This report shall summarize the impact that these results had on the Application Reference Model and the final scope of the Application Protocol. This report may include an issues log. This report shall be included in the Application Protocol validation report.

The validation of the Application Reference Model is a critical and resource intensive activity. Complete model validation of a complex Application Reference Model is impractical. It is usually evident from the development of the Application Activity Model that recurring demands for the same information exist. These facts can be used to prioritize the parts of the Application Reference Model to be validated. The objective of the Application Reference Model validation testing is to provide a significant level of confidence in the correctness and robustness of the model. The Application Protocol team shall document the required level of confidence and the steps that were completed to reach that level.

| Stuart Lord comment - translate as “it's impossible to get the requirements right!” Conflicts with statement page 46 “Personnel shall be assigned to work with the AP project to ensure the model is correct and complete." |
|---|

One method for validating an Application Reference Model is to build a prototype database which closely matches, if not replicates, the constructs of the Application Reference Model. This prototype database is then tested for its ability to accommodate representative test parts or products from the application context. Representative usage tests in the form of queries are posed upon these populations of the database to evaluate whether the “simulated Application Reference Model” is sufficient to support the in-scope processes defined in the Application Activity Model.

Application Reference Model validation shall, at a minimum, include paper populations of the data structure and reference path analyses to check whether the Application Reference Model can support the representative test parts and can support the representative usage tests. Both of these methods will benefit from the use of software tools to control and aid in the development and testing of complex Application Reference Models.

Detailed validation testing of the Application Reference Model provides feedback on the structure and requirements defined in the model. There will be iterations between the population and reference path analysis of the Application Reference Model and its development. Each iteration shall be documented in the Application Reference Model validation report along with a summary of the test coverage, data, and path analyses which were used in the validation, and a summary of the impact of the test results on the model.

## 9.6. Application Interpreted Model

Assigned to: MEG/JPF

This text taken from APG 4.4 Development and review of the application interpreted model

The Application Interpreted Model (AIM) is an EXPRESS schema which specifies the interpretation of the integrated resources to satisfy the information requirements of the Application Protocol. The Application Interpreted Model specifies the implementable constructs of the Application Protocol. The Application Interpreted Model Development Project, with the assistance of the Application Protocol team, shall produce the Application Interpreted Model. The documentation for an Application Interpreted Model includes six components.

1. Mapping table
2. Application Interpreted Model EXPRESS short listing
3. Application Interpreted Model EXPRESS annotated listing
4. Application Interpreted Model EXPRESS-G
5. Integrated resources interpretation report
6. Application Interpreted Model validation report

### 9.6.1. Integrated resources interpretation

Text taken from APG 4.4.1 Integrated resources interpretation

The Application Interpreted Model is developed by interpreting the integrated resource constructs based on the information requirements. The Application Interpreted Model may subtype or add global rules to integrated resource constructs in order to satisfy those application specific requirements which are specified in the Application Reference Model. Since the integrated resources are generic in nature, Application Interpreted Models will take a more specific view of these entities. To restrict the population of particular attributes of an entity, a global rule shall be specified. This rule shall contain constraints on the particular entity or group of related entities.

Subtyping of an integrated resource construct is necessary when an Application Reference Model concept is only partially supported by that construct. In this case, a subtype of that integrated resource construct shall be created and those attributes and rules necessary to complete the concept attached to that subtype. Only DERIVED attributes may be specified in the Application Interpreted Model except for the purpose of completion and assignment of management resources. Subtyping in the Application Interpreted Model is also used to specify the constructs in the Application Interpreted Model to which product data management resources (such as date_time, person_organization, approval, etc.) shall be applied. In this case, a SELECT type shall be created in the Application Interpreted Model which specifies the entities which are required to have product data management resources applied to them. A subtype of the entity in the specific product data management schema shall be created and an attribute placed in that entity whose type is a SET [1:?] of the newly created SELECT.

The Application Protocol team shall document the rationale for each subtype and rule added in the Application Interpreted Model in an integrated resources interpretation report. This report shall be included in the Application Protocol validation report. The Application Interpreted Model Development Project shall manage the development of proposed subtypes.

As the number of Application Protocols grows over time, there will inevitably be some overlap in scope between different Application Protocols. These areas of overlap may be indicated by commonality in the scope statements or the use of the similar UoFs in their Application Reference Models. These areas may also be indicated by the common use of resource constructs within Application Protocols. When two Application Protocols contain equivalent information requirements, these Application Protocols shall use the same interpretation of the integrated resource constructs. Figure 8-1 depicts this aspect of the Application Interpreted Model development and integration process.

When interpreting the integrated resources to match the information requirements, the following steps shall be taken[14]:

#### 9.6.1.1. Identify integrated resource constructs corresponding to application objects.

To represent the required functionality using integrated resource constructs, each of the application objects must be examined to find a corresponding construct or group of constructs in the integrated resources. At this point of interpretation, only those constructs which satisfy an information requirement directly shall be identified. The method by which the integrated resources are interpreted directly by the Application Interpreted Model is described in “Application Interpreted Model EXPRESS documentation” later in this section.

#### 9.6.1.2. Identify requirements for specializing integrated resource constructs.

In addition to those constructs in the Application Reference Model which have a direct correspondence to constructs in the integrated resources, there may be some constructs which have a partial correspondence. Some constructs in an Application Reference Model may correspond to the general semantic intent of an integrated resource construct, yet require the extension or constraint of that construct to attain complete semantic correspondence. The completion of these concepts is called specialization. At this point in the Application Interpreted Model development process, those constructs requiring specialization in the Application Interpreted Model are identified.

[14] See STEP Development Methods: Resource Integration and Application Interpretation, NISTIR, National Institute of Standards and Technology, Draft, March 1992 and Guidelines for AIM Development, ISO TC184/SC4/WG4(P3) Document N302, September 1993.

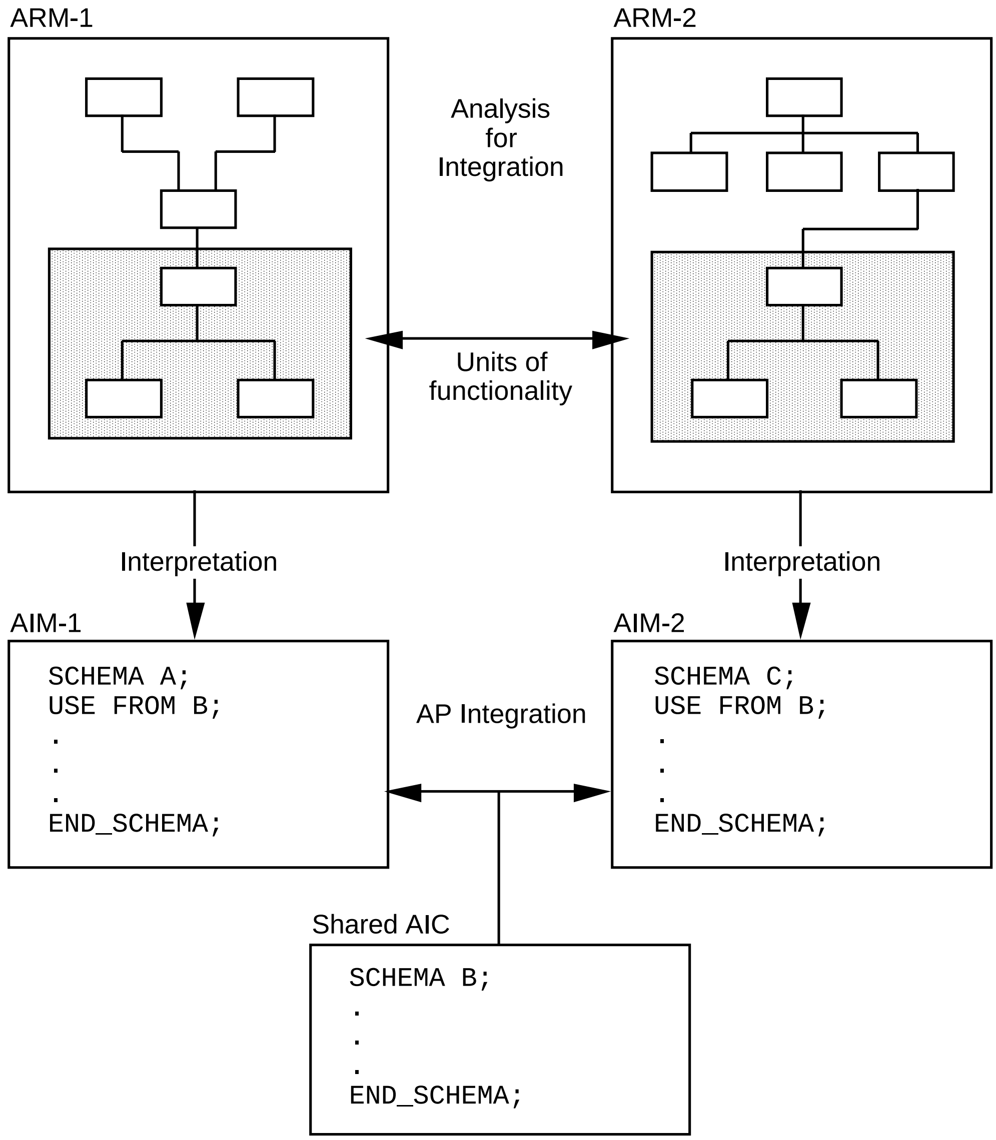


Figure 4: Analysis for Application Protocol integration

### 9.6.1.3. Specialize constructs with partial correspondence.

For the integrated resources which need specialization in order to completely satisfy information requirements, subtypes with attributes, rules, and constraints are added. There are two types of specialization which may be done in an Application Interpreted Model. The first type of specialization defines a new subtype to match the requirements of the context. This is accomplished by defining a subtype of the integrated resource entity whose semantics are being refined. The subtype entity contains DERIVED attributes with attribute names that are specialized to be consistent with the semantics of the information requirements. The second type of specialization is the additional constraining of an entity from the integrated resources. When a new constraint needs to be added to the integrated resource in order to satisfy the information requirements, either a local or a global rule shall be added to the Application Interpreted Model depending on the use of the construct. See “Guidelines for Application Interpreted Model Development”, ISO TC184/SC4/WG4(P3), document N302, September 1993, for the specific rules on developing the application interpreted model.

SPL: rather than referencing the WG4 document, should this material be included here?

#### 9.6.1.4. Develop mapping table

While the Application Interpreted Model is developed, the Application Protocol team shall record in a table in clause 5.1 of the Application Protocol document the selections and specializations made from the integrated resources to establish correspondence to each information requirement. This mapping will list each information requirement and its corresponding Application Interpreted Model construct(s). If a path of entity references in the Application Interpreted Model must be followed to completely satisfy a particular requirement as it is given in the Application Reference Model, the mapping table shall indicate the complete reference path which needs to be given to represent the required information in the Application Interpreted Model.

#### 9.6.1.5. Develop integrated resource interpretation report.

The Application Protocol team shall produce an integrated resource interpretation report to summarize the rationale with which the Application Interpreted Model was derived and all specializations of integrated resource constructs. In developing the constraints on entities, a number of constraints may be required to constrain different entities for the same purpose. These constraints shall be grouped together in the integrated resource interpretation report with a description of their purpose in the Application Interpreted Model.

### 9.6.2. Mapping table

| Text taken from: APG 4.4.2 Mapping table |
|---|

During the interpretation of the integrated resources, the mapping of the correspondence between the application objects and the constructs of the Application Interpreted Model shall be documented and maintained. The resultant mapping table shows the Application Interpreted Model construct(s) required for each application object.

In EXPRESS, a single attribute or entity may not be enough to establish a full understanding of a concept. Although a single attribute or entity may be the Application Interpreted Model construct to which an Application Reference Model construct maps, that attribute or entity will not provide all of the information necessary to completely understand the semantic. The specification of a reference path in the mapping table occurs when an attribute in the Application Reference Model and the entity to which it belongs do not correspond to the same entity in the Application Interpreted Model. Often an attribute in the Application Reference Model is developed at a higher level of detail than the integrated resources. In this case, the reference path is provided so that the complete semantic (including the relationship of the attribute to the entity in the Application Reference Model) is represented in the mapping table. Additionally relationships in the Application Reference Model will always have reference paths to show the complete set of entity instances required in the Application Interpreted Model to satisfy the relationship, subtypes created in the Application Interpreted Model will show the supertype from the integrated resources in the reference path and any mapping rules or choices will be specified in the reference path.

The example table below (see Figure 4-5) illustrates a number of types of mappings that will be found in the mapping table of an application protocol. Two units of functionality are given, Advanced_b_rep and Authorization. The mappings of two application elements, ADVANCED_B_REP and APPROVAL, are provided. The mappings are described as shown in table 2.

Table 2: mapping table example

| **Application element** | **Application Interpreted Model element** | **Source** | **Rules** | **Reference path** |
|---|---|---|---|---|
| **Advanced_b_rep** | | | | |
| ADVANCED_B_REP | advanced_brep_representation | 203 | | shape_representation => advanced_brep_representation |
| **Authorization** | | | | |
| APPROVAL | cc_design_approval | 203 | 1,2 | approval_assignment => cc_design_approval |
| date | date | 41 | | cc_design_approval<= approval_assignment approval_assignment.assigned_approval-> approval<- approval_date_time.dated_approval approval_date_time.date_time-> date_time_select=date_and_time date_and_time.date_offset-> date |
| purpose | approval.purpose | 41 | | cc_design_approval <= approval_assignment approval_assignment.assigned_approval -> approval approval.purpose |

1. approval_requires_approval_date_time

2. approval_requires_approval_person_organization

- The application element ADVANCED_B_REP maps to the Application Interpreted Model entity advanced_brep_representation. The source column value denotes that the Application Interpreted Model entity advanced_brep_representation is an Application Protocol specialization, originating in AP 203. This specialization requires a reference path from the integrated resource entity from which it is specialized. The reference path denotes that the Application Interpreted Model entity advanced_brep_representation is a subtype of the integrated resource entity shape_representation.

- The application element APPROVAL maps to the Application Interpreted Model entity cc_design_approval. The source column denotes that the Application Interpreted Model entity cc_design_approval originates in AP 203. This specialization requires a reference path from the integrated resource entity to the specialized subtype. Rules 1 and 2 which are found at the end of the table constrain the use of the approval structure.

- The application element APPROVAL has an attribute date which maps to the date entity in the Application Interpreted Model. The date entity originates in Part 41 as indicated in the source column. Since the attribute maps to an entity in the Application Interpreted Model, a reference path is give from the entity cc_design_approval (this is the entity to which the application element APPROVAL was mapped) to the date entity (this is the entity to which the Application Reference Model attribute date is mapped). The reference path is to be read as follows:

    - cc_design_approval is a subtype of approval assignment,

- approval_assignment has an attribute named assigned_approval that references the entity approval,
- approval is referenced by the attribute dated_approval in the entity approval_date_time,
- approval_date_time has an attribute named date_time which references a select type called date_time_select,
- in this case, the date_time_select, references the date_and_time entity,
- the date_and_time entity has an attribute named date_offset,
- the attribute date_offset references the entity date.

- The application element APPROVAL has an attribute purpose which maps to the purpose attribute of the approval entity in the Application Interpreted Model. The source of the attribute purpose in the entity approval is Part 41.

ISO 10303 experts and application experts shall review the mapping table to ensure that they are complete and correct. The mapping is complete when each Application Reference Model construct has an equivalent construct(s) in the Application Interpreted Model.

### 9.6.3. Application Interpreted Model EXPRESS documentation

Text taken from: Guidelines to the development of ISO 10303 Application Protocols 4.4.4 Application Interpreted Model EXPRESS documentation/

The Application Interpreted Model shall be documented in three formats:

- Application Interpreted Model EXPRESS-G
- Application Interpreted Model EXPRESS short listing
- Application Interpreted Model EXPRESS annotated listing

#### 9.6.3.1. Application Interpreted Model EXPRESS-G

Text taken from: Application Protocol Guidelines 4.4.4.1 Application Interpreted Model EXPRESS-G

Once an initial correspondence between application elements and the Integrated resources constructs has been established, an EXPRESS-G representation of the Application Interpreted Model should be produced. The EXPRESS-G diagrams are useful in developing the Application Interpreted Model EXPRESS listings, both the short listing and the annotated listing. The Application Interpreted Model EXPRESS-G diagrams shall include all ENTITYs and ENUMERATION and SELECT types. An EXPRESS-G model for all entities in the Application Interpreted Model shall be provided as annex G.

#### 9.6.3.2. Application Interpreted Model EXPRESS short listing

Text taken from: Application Protocol Guidelines 4.4.4.2 Application Interpreted Model EXPRESS short listing

The Application Interpreted Model EXPRESS short listing provides the interface specification between Application Interpreted Model schema and the resources it uses, i.e., integrated resources and Application Interpreted Constructs. There is a single EXPRESS schema for each Application

Interpreted Model. This EXPRESS schema specifies the elements from the integrated resources and the Application Interpreted Constructs that are used in the Application Protocol and contains the types, entity specialisation, rules, and functions that are specific to the Application Protocol. All entities from the integrated resources, including those which are to be specialised, shall be specified in the Application Interpreted Model EXPRESS short listing using the USE FROM construct.

Any entity which is declared in the Application Interpreted Model EXPRESS short listing shall be a subtype of an entity that is brought into the scope of the schema from using the USE FROM statement. These are the integrated resources entities which need to be specialised in the Application Interpreted Model. Data types shall be specified in the short listing to specialise the used integrated resources. Rules and functions which are needed to further constrain an entity or relationship shall be included in the short listing.

The Application Interpreted Constructs shall be incorporated into the Application Interpreted Model schema by the use of the USE FROM <aic_schema> statement. Each Application Interpreted Constructs USE FROM statement shall be followed by a tail comment which references the subclause of annex A in which the Application Interpreted Construct EXPRESS schema is specified. An Application Interpreted Construct only shall be used in an Application Interpreted Model in its entirety.

### 9.6.3.3. Application Interpreted Model EXPRESS annotated listing

Text taken from: Application Protocol Guidelines 4.4.4.3 Application Interpreted Model EXPRESS annotated listing

The Application Interpreted Model EXPRESS annotated listing consists of the complete documentation of the expanded USE FROM statements and the Application Protocol unique EXPRESS declarations. The annotated listing shall contain definitions of each type, entity, attribute, rule, and function in the Application Interpreted Model schema. For each type, entity subtype, rule, and function that is defined in the Application Interpreted Model EXPRESS short listing, the description in the Application Interpreted Model EXPRESS annotated listing shall reference the specification in the corresponding subclause of the Application Interpreted Model EXPRESS short listing and not repeat the definition. Discrepancy between the short listing and the annotated listing shall not occur.

The Application Interpreted Model EXPRESS annotated listing shall contain a schema which defines the Application Interpreted Model, along with the Application Interpreted Construct schema which are incorporated into the Application Interpreted Model. The fundamental concepts, scope, and assumptions for each Application Interpreted Construct shall be documented with the Application Interpreted Construct schema.

## 9.6.4. Application Interpreted Model validation

Text taken from: Application Protocol Guidelines 4.4.5 Application Interpreted Model validation

Comprehensive validation testing of a complex Application Interpreted Model is resource intensive. The objective of Application Interpreted Model validation testing is to provide a significant level of confidence in the correctness and robustness of the model. The Application Protocol team shall document the required level of confidence and the steps that were completed to reach that state.

The Application Protocol team shall summarise the Application Interpreted Model validation test plan and test results in an Application Interpreted Model validation report. This report shall include the rationale for the selection of test purposes, test pieces, and usage scenarios. A selection of the Application Reference Model validation tests shall be included in the Application Interpreted Model tests. An analysis of the degree of coverage provided by the validation testing shall be included. This report may include an issues log. This report shall be included in the Application Protocol validation report.

## 9.7. Application interpreted constructs

Assigned to: MEG/JPF

Text taken from: APG 4.4.3 Application interpreted constructs

During the interpretation process, one or more groups of constructs may be identified for which the information requirements are equivalent to another ISO 10303 application protocol. In this case, common interpretations are used and schema to satisfy the common information requirements are developed for inclusion in the application interpreted models. These schema are referred to as application interpreted constructs. All application protocols which have been identified as sharing a common information requirement shall use the appropriate application interpreted construct schema in its entirety. The Application Protocol Integration Project of WG4 will keep a library of all application interpreted constructs and a list of all application interpreted models of ISO 10303 application protocols which use them.

### 9.7.1. Purpose of Application Interpreted Constructs

Text taken from: Chia-hui Shih Guidelines for Application Interpreted Construct development WG5 N112, 2.3 AIC purpose

The main purpose of an Application Interpreted Construct is to provide a mechanism to identify and encapsulate the common requirements of distinct applications represented by their respective Application Reference Models. When the need for a common application requirement is identified, an application interpretation process, the same as what is used in Application Interpreted Model development, will take place to develop a new Application Interpreted Construct. Once the Application Interpreted Constructs are identified, the shareability of the Application Interpreted Construct specification among different Application Protocols become explicit, whereas without Application Interpreted Construct, all the common requirements are buried in the populated long forms of the Application Interpreted Models.

The benefit of the explicit sharing of data definitions for the common requirements among Application Protocols, is more than eliminating redundancy (multiple copies of the same data among different applications); it may improve the communication among Application Protocol implementations on their common requirements based on common data. Application Protocol interoperability is defined to require three levels of sharing: data definition sharing, data instance sharing, and processor sharing. The use of Application Interpreted Construct does not directly provide all three levels of interoperability sharing, however, it does provide the mechanism for explicit data definition sharing and through which, data instance sharing and processor sharing can be facilitated. Because Application Interpreted Constructs are modules of application interpreted constructs, the use of Application Interpreted Construct allows one to build bigger applications over smaller applications with well maintained history trees both at the conceptual level (EXPRESS model) and the instance level (physical file, ISO 10303-22, databases).

### 9.7.2. Application Interpreted Construct scoping and content

Text taken from: Chia-hui Shih Guidelines for AIC development WG5 N112, 2.4 AIC scoping and content

A group of integrated resource entities or constructs may be used by multiple Application Interpreted Models but they are not necessarily Application Interpreted Constructs. The primary difference of Application Interpreted Construct and a group of integrated resource entities that are shared by multiple Application Interpreted Models is that the semantics of the group must provide a specific requirement shared by Application Protocols, such as a shell_based wireframe shape representation or draughting annotation. A key element of the strategy for the development, use, documentation, and management of Application Interpreted Constructs is to provide common interpreted constructs that meet the requirements common to multiple Application Protocols. The specification of requirements well defines the semantic boundary of an Application Interpreted Construct, lending it stability, and moreover, controls the number of Application Interpreted Constructs.

The content of an Application Interpreted Construct must at a minimum include the following elements:

1. Each Application Interpreted Construct is specified with at least one root entity, each of which is always a SUBTYPE of an integrated resource entity.

2. No global rules are allowed in an Application Interpreted Construct, all constraints which apply globally to the elements of an Application Interpreted Construct are specified as local rules in the Application Interpreted Construct root entity.

3. Each Application Interpreted Construct root entity will define an attribute "aic_function" to identify the unique requirement (data content) each Application Interpreted Construct provides and the Application Interpreted Construct schema name. All root entities within a single Application Interpreted Construct will have the same functional context, specified in EXPRESS constructs that follow the template provided. The functional contexts are unique across all ISO 10303 Application Interpreted Construct's.

4. The name of the constant shall be aic_xxx_fd, where aic_xxx is the schema name of the Application Interpreted Construct.

The following EXPRESS example illustrates the constructs found in an Application Interpreted Construct.

```
*)

SCHEMA aic_xxx;

USE FROM ir_schema;

CONSTANT
   aic_xxx_fd : LIST [2:2] OF STRING;
              : = ['AIC_XXX', 'AIC FUNCTION DESCRIPTION'];
END_CONSTANT;

ENTITY aic_a;
SUBTYPE OF (ir_a);
   aic_function  :  LIST [2:2] OF STRING;
WHERE
   WR1 : aic_function = function_definition;
   WR2 : -- aic specific constraints on a
END_ENTITY;

END_SCHEMA;

(*
```

### 9.7.3. Application Interpreted Construct usage in an Application Interpreted Model

Text taken from: Chia-hui Shih Guidelines for AIC development WG5 N112, 2.5 AIC usage in an Application Interpreted Model

The use of an Application Interpreted Construct is specified within the EXPRESS “USE FROM” keyword, followed by the Application Interpreted Construct schema name. The use of Application Interpreted Construct must be of its entirety. However, rules may be written to constrain the usage of Application Interpreted Constructs within an Application Interpreted Model. When an Application Interpreted Construct is used in an Application Interpreted Model, domain constraints may be added in the using Application Interpreted Model that governs the valid population of the Application Interpreted Construct constructs. When more than one Application Interpreted Construct is used in an Application Interpreted Model, global rules may be written to constrain the relationships amongst these Application Interpreted Constructs.

### 9.7.4. Application Interpreted Construct development procedure

Text taken from: WG5 N112, Chia-hui Shih, Yuhwei Yang, AIC Guidelines, 3. Development procedure

Since Application Interpreted Construct development is a part of the Application Interpreted Model development, the guidelines for Application Interpreted Construct development are consistent with that of the Application Interpreted Model development.

At the onset of an Application Protocol development, the available ISO 10303 resources for the developers are the Integrated Resources and existing Application Interpreted Constructs. However, at the stage of developing the Application Reference Model, it is not a requirement for an applications expert to have an intimate knowledge of the ISO 10303 integrated resources or Application Interpreted Constructs or to be proficient on EXPRESS as an information modelling language. Hence the development of an Application Interpreted Model (AIM) requires two types of expertise in close interaction: in the field of the application and in ISO 10303 Integration. This is a labour intensive process. It will certain shorten the development cycle if the application experts are also ISO 10303 integration experts.

The following is the development phase outline:

#### Identify application requirements

| | |
|---|---|
| Task: | identification of context and scope for the application and creation of Application Activity Model and Application Reference Model |
| Tools/Background: | using either one of the following information modelling languages: IDEF1X, NIAM, or EXPRESS. ISO 10303 resource model knowledge is not required. |
| People: | the experts in the application domain |
| To deliver: | Application Protocol group 1 document which includes an information model (Application Reference Model), represented by a popular language in the application domain that is ready to be interpreted. |

## Interpretation of application requirements

Task: Semantic analysis of Application Reference Model model with the ISO 10303 integration

Tools/background: ISO 10303 schema resources, including Integrated Resources, existing Application Interpreted Constructs, Application Interpreted Models and integration methodology

People: Applications experts and ISO 10303 integrators

To deliver:

A: identify in the Application Reference Model those requirements that are already represented by some existing Application Interpreted Constructs.

B: identify requirements in the Application Reference Model that are common with another Application Reference Model but are not already represented by an Application Interpreted Construct.

C: preliminary proposal for new Application Interpreted Constructs.

## Interpretation phase

Task: Selection of Integrated Resources, Application Interpreted Constructs that support the Application Reference Model and identifying new Application Interpreted Constructs

Tools/background: ISO 10303 schema resources, including Integrated Resources, Application Interpreted Constructs, Application Interpreted Models and integration methodology

People: applications experts, ISO 10303 integrators

To deliver:

A: Integrated Resources that satisfy Application Reference Model requirement

B: additional EXPRESS constructs that specialises the selected Integrated Resources to satisfy the Application Reference Model requirements

C: proposal for new Application Interpreted Construct and its scope

D: mapping of application objects to the existing Application Interpreted Constructs and new Application Interpreted Constructs.

## Development of new Application Interpreted Constructs

Task: Determining the content of new Application Interpreted Constructs

Tools/background: ISO 10303 resources, EXPRESS tools

People: ISO 10303 integration and EXPRESS

To deliver: The detailed content of the new Application Interpreted Constructs including the following:

- Scope - description of the Application Interpreted Construct scope in English
- Short form - identify the elements in the Application Interpreted Constructs short form
- Determining USE FROM statements to pull in Integrated Resource entities within scope, to cover SUBTYPES etc.
- Copying Application Protocol specific constructs
- Identifying root node(s)
- Identifying Application Protocol specific global rules on Application Protocol specific entities within Application Interpreted Construct and rewrite as local rules in root node
- Long form - obtaining it by compiling short form

### Preliminary review

### Documentation

All newly identified Application Interpreted Constructs will be given a name of aic_xxx. A separate document is prepared according to the Application Interpreted Construct document guidelines for each new Application Interpreted Construct.

### Distribution/Review/Approval/Archiving

Task: Register the Application Interpreted Construct document (at each stage of development cycle) to the Application Interpreted Construct library for document management and configuration control.

## 9.8. Application Protocol usage guide

| Text taken from: Application Protocol Guidelines 4.4.6 Application Protocol usage guide |
|---|

During the development and validation of the Application Protocol, descriptions of the way in which the Application Protocol is to be used and illustrative business cases shall be developed. This information shall be continually refined with the objective of eventually producing an Application Protocol usage guide. The Application Protocol usage guide is an optional and informative annex of the Application Protocol.

A subclause in the usage guide may be reserved for each applicable implementation method. Examples of the use of the Application Protocol within each implementation method applicable may be explained in these subclauses. For a file exchange implementation, an example exchange file may be included. The inclusion of an exchange structure with values which are representative of those for which the Application Protocol was specified is an appropriate element of an Application Protocol usage guide.

## 9.9. Conformance classes

| Assigned to: JPF |
|---|

Text taken from: Application Protocol Guidelines 4.5 Development and review of conformance requirements

Conformance testing is the evaluation of an implementation for all required characteristics, i.e., to determine whether an implementation conforms to the standard. For an Application Protocol, this includes the information requirements, the Application Interpreted Model (entities, types, attributes, functions, procedures, rules, and the full range of values), and any implementation specific requirements defined in annex C of the Application Protocol.

JPF: 6/6/95: the rest of this clause includes references to specific SC4 working groups and projects. As stated in the Scope, it is the intention of this Reference Manual to eliminate such references. They are left here pending discussion with the PPC and others of the eventual home for documentation of the SC4 procedures for ISO 10303 development (as opposed to the procedures to be applied independent of organisation, which are the subject of this document).

With the assistance of WG6, the Application Protocol team shall define the conformance requirements and any conformance classes[15] for the Application Protocol after careful analysis of the requirements of industry, the objectives of conformance testing, and the consequences of enforcing or not enforcing completeness. If conformance classes are used, the conformance requirements for the individual classes shall be explicitly listed in clause 6.

If completeness is enforced, each implementation must be able to process the full range of values for every attribute of every entity identified in a particular conformance class. For example, if widgets have a colour attribute which is an enumeration of red, blue, and green, then all conforming processors must be able to produce (pre-processor) or interpret (post-processor) all three colours of widgets. This does not imply that all three colour of widgets must appear in each exchange, merely that a system can distinguish widgets of the different colours, if desired.

Using the above example, if completeness of colour attributes is not enforced, a “conforming” implementation may be at liberty to lose all the colour information or to change blue and green widgets into red ones. From the perspective of the industrial need to maintain the colour information, this condition is unacceptable. Application Protocol developers are advised to require completeness of implementation of all Application Protocol required characteristics. A clear statement of the required completeness shall be included in clause 6 of the Application Protocol.

A list of high level conformance requirements for an Application Protocol implementation is provided below. The Application Protocol developers shall consult WG6 on proposed modifications and additions to these conformance requirements.

- The information requirements of the Application Protocol shall be preserved in the implementation. This includes support for valid combinations of entities and their attributes. Consequently, all application objects and assertions from clause 4 shall be maintained.

- All entities, types, and their associated constraints identified in a particular conformance class shall be supported. Treatment of options and default values shall conform to the Application Interpreted Model.

- Only those constructs specified in the Application Interpreted Model shall be produced or accepted by an implementation.

- An implementation of ISO 10303 combines an application protocol with an implementation form. Such an implementation shall satisfy all general requirements applicable to the

[15] The definition of a conformance class shall include a table indicating which constructs from the AIM shall appear in each conformance class.

implementation form (given in the appropriate part of the 20-series class) and any Application Protocol-specific options given in annex C of the Application Protocol.

Clause 6 of an Application Protocol shall reference the implementation specific requirements specified in annex D of that Application Protocol. This reference includes the correct use of the Application Interpreted Model mapped by the implementation form and conformance to the implementation form.

The conformance requirements shall be reviewed and evaluated by experts (application experts, implementation experts, and Application Protocol methods experts) who did not participate in the original development. These experts and WG6 shall assess the utility, practicality, understandability, and coverage provided by this clause of the Application Protocol. The results of this review shall be included in the Application Protocol validation report.

## 9.10. Summary of application protocol validation

Text taken from: Application Protocol Guidelines 4.6 Summary of application protocol validation

The basic concept of Application Protocol validation is to ensure that the scope and information requirements are completely and unambiguously delivered in the Application Protocol. This requires that the scope, requirements, Application Reference Model, Application Interpreted Model, and conformance requirements are complete and consistent. The components of the Application Protocol shall be evaluated for their soundness and for their internal consistency. Each Application Protocol project shall develop and ensure the execution of an Application Protocol validation plan. The Application Protocol validation plan shall be reviewed with the WG4 Application Protocol Guidelines and Framework Project and WG6.

The Application Protocol validation plan and the resulting Application Protocol validation report shall be maintained by the Application Protocol project as a supporting document during the development of the Application Protocol. A well documented Application Protocol validation report is a useful reference for the Application Protocol project and reviewers of the Application Protocol while the Application Protocol is being developed. The completed Application Protocol validation report shall be submitted with the completed Draft Application Protocol for review and acceptance by the WG4 Qualification Project.

As part of the validation process it is essential to include the viewpoints of many individuals from the different disciplines that may use the Application Protocol. Suggested reviewers of the Application Protocol include:

- application experts to determine whether the Application Protocol comprehensively and unambiguously describes the application domain;
- experts in the modelling methodology used for the Application Reference Model to determine that the Application Reference Model is correctly specified;
- ISO 10303 experts to determine that the Application Protocol correctly uses the ISO 10303 specification;
- implementation experts to determine the utility and implementability of the Application Interpreted Model and any specified conformance classes.

A summary of the Application Protocol validation process for proposed Application Protocols is given below, followed by a more detailed description of the complete methodology:

1. Scope and requirements evaluation analyses the completeness and correctness of the scope, requirements, and Application Activity Model.

2. Application Reference Model validation evaluates the completeness and correctness of the Application Reference Models representation of the information requirements for the application area and correspondence to the scope and Application Activity Model.

3. Application Interpreted Model validation evaluates the completeness and correctness of the Application Interpreted Model's representation of the Application Protocol information requirements as specified by the Application Reference Model.

4. Conformance requirements evaluation analyses the completeness of coverage, correctness, and self-consistency of these requirements with the Application Reference Model, Application Interpreted Model, and implementation requirements.

5. Application Protocol validation through simulated implementations via prototypes uses test implementations of the Application Protocol to evaluate the utility, correctness, and completeness of the Application Protocol.

Further details are given in the following sections.

### 9.10.1. Scope and requirements evaluation

Activity 1, scope and requirements evaluation, requires a team of experts from the subject application area to provide peer reviews of the scope, requirements, and Application Activity Model. The Application Protocol team shall conduct a walk-through and evaluation of these items by these experts. The majority of these expert reviewers should not have participated in the Application Protocol modelling effort. The objectives of this review are to ensure that:

- the scope and requirements are accurate, viable, and complete to meet an important industrial need;

- the Application Activity Model accurately represents all areas which are described by the refined scope statement, and

- the documentation is sufficient, unambiguous, and conveys the correct meanings.

Sample instances of the concepts that the Application Protocol is intended to support are also used to verify the scope and requirements statement.

### 9.10.2. Application Reference Model validation

Activity 2, Application Reference Model validation, ensures that the Application Reference Model satisfies the stated scope and requirements of the Application Protocol, is consistent with the Application Activity Model, and is syntactically and semantically complete and correct. This activity uses experts from the subject application area to provide peer reviews of the Application Reference Model. The sample test parts or products and the usage scenarios which were used to define the scope should be used to validate the Application Reference Model.

For an optimum model validation of the Application Reference Model, the reviewers should not be the same experts that participated in the development of the Application Reference Model. This portion of the process is manpower intensive. In the future it may be possible to use software tools to evaluate the Application Reference Model for completeness or correctness.

The Application Reference Model shall be reviewed manually to establish the validity of the semantics of the Application Reference Model. The Application Reference Model shall be reviewed to verify that the relationships between application objects are understandable and correct and that all required objects appear in the Application Reference Model. In addition, the Application Reference

Model should be loaded into a software tool(s) to check consistency of the Application Reference Model and the completeness of the representation.

The definitions of all application objects and relationships shall be checked for completeness and understandability. The definitions shall be understandable to the prospective users of the Application Protocol. The users of the Application Protocol will include developers of ISO 10303 implementations and application experts. There shall be a definition for every element in the Application Reference Model. Some Application Protocol developers have been tempted to omit definitions of certain elements on the grounds that they are “self evident” or “standard terms”. The problem with this approach is that what is self evident to one individual is not self evident to another. When this model validation is successfully completed, the Application Protocol team shall produce a summary report on the Application Reference Model validation.

### 9.10.3. Application Interpreted Model validation

Activity 3, Application Interpreted Model validation, involves the evaluation of the Application Interpreted Model and the specified implementation form(s) for the ability to carry all of the information requirements specified in the Application Protocol. This model validation shall check that all items of information defined in the Application Reference Model can be carried in the Application Protocol format as specified by the Application Interpreted Model and any implementation specific requirements. The objective is to ensure semantic correspondence between the Application Reference Model and the Application Interpreted Model. The Application Reference Model and the Application Interpreted Model shall be checked to verify that they truly correspond to each other. They shall be checked two ways:

- all Application Reference Model constructs map completely to one or many Application Interpreted Model constructs, and

- all Application Interpreted Model constructs map completely to one or many Application Reference Model construct.

An important aspect to check is that all the constraints modelled in the Application Reference Model are represented in the Application Interpreted Model. The Application Interpreted Model shall be successfully compiled on the EXPRESS compiler(s) designated by the Qualification and Validation Project. The results of the compilation(s) of the Application Interpreted Model shall be included in the Application Interpreted Model validation report. Application Interpreted Model validation shall require both application area experts and experts in the capabilities and use of ISO 10303 to generate populated test pieces and path traversals.

### 9.10.4. Conformance requirements evaluation

Activity 4, conformance requirements evaluation, analyses the completeness of coverage, correctness, and consistency of these requirements with the Application Reference Model, Application Interpreted Model, and implementation requirements. The conformance requirements shall be checked to ensure that they meet industry needs and are useful, testable, and stated clearly. If conformance classes are also specified for the Application Protocol, the conformance classes are evaluated against these same criteria.

### 9.10.5. Application Protocol validation

Activity 5, Application Protocol validation by developing and testing Application Protocol prototypes, is a recommended additional step for providing a higher level of confidence in the implementability and utility of the Application Protocol. The creation of a prototype requires Application Protocol developers to analyse the Application Protocol in ways that probably would not be considered in reviews of the Application Protocol. If a prototype is not developed it is essential that the Application Protocol validation efforts include the development of detailed usage scenarios and test cases to simulate validation checks that a prototype software development effort would produce.

The development and validation of a ISO 10303 Application Protocol is an iterative process of progressive detailing and refinement. Each step in this process provides critical feedback for the next version of the draft Application Protocol. While the Application Protocol validation effort is underway, new versions of the Application Protocol may be released during the review process. It is important to ensure that comments generated on previous versions of an Application Protocol are addressed by the new version of the Application Protocol.

### 9.10.6. Application Protocol validation report

Text taken from: Application Protocol Guidelines 4.6.1 Application Protocol validation report

The Application Protocol validation report documents: the Application Protocol validation process, the results of each validation activity, how these results were evaluated, and how all validation problems and errors were resolved. The Application Protocol validation report shall be developed in parallel with the development of the Application Protocol. The report shall include the following clauses:

1. Introduction
2. Application Protocol validation plan
3. Scope and requirements evaluation report
4. Application Reference Model validation report
5. Integrated resources interpretation report
6. Application Interpreted Model validation report
7. Conformance requirements evaluation report

The Application Protocol validation report may include an additional clause on Application Protocol validation with prototype implementations and an annex describing sample test parts, usage scenarios, and success criteria used during the Application Protocol validation activities.

Clause 1 provides an overview of the validation process, problems identified and procedures used to resolve problems, issues, and errors. The Introduction shall also describe market assessments of industry needs for the Application Protocol and any metrics used to assess industry and application experts confidence in the results of the validation activities.

Clause 2 describes the Application Protocol validation plan and the resources and experts used to complete the validation. This clause may also document the relationships between each validation activity and how traceability between activities and results was maintained. Collaboration on validation activities with other ISO 10303 projects shall be documented in this clause.

Clause 3 describes the analysis performed to determine the accuracy and completeness of the Application Protocol's scope, functional requirements, and Application Activity Model. This analysis is based on the requirements from industry and the evaluations by industry and application experts.

Clause 4 describes the validation testing of the Application Reference Model. These tests verify the completeness and correctness of the information requirements modelled in the Application Reference Model and the Application Reference Model's correspondence to the scope and Application Activity

Model. This clause shall document that each “in scope” information stream of the Application Activity Model is supported by the Application Reference Model.

Clause 5 records the interpretation of the integrated resources to meet the information requirements of the Application Protocol and the resolutions to problems identified during the interpretation process. This clause shall include sections of the Application Protocol's mapping tables and explanations necessary to clarify the rationale for selections from the integrated resources and modifications to the integrated resources.

Clause 6 describes the validation testing of the Application Interpreted Model. This validation ensures the correlation of the information requirements in the Application Reference Model to the constructs defined in the Application Interpreted Model and verifies that the Application Interpreted Model is complete and self-consistent. The results of the compilation(s) of the Application Interpreted Model shall be included in this clause.

Clause 7 describes the assessment of the completeness of coverage, correctness, and self-consistency of the conformance requirements and any specified conformance classes with the information requirements, Application Interpreted Model, and implementation requirements.

During the development of the Application Protocol, the Application Protocol project may choose to distribute a subset of these clauses to selected reviewers. The Application Protocol validation report, with all relevant clauses completed, shall be submitted with the Application Protocol as part of each review by the WG4 ISO 10303 Part Qualification and Validation Project.

## 9.11. Abstract test suite development

Assigned to: SPL

JPF: the method as described here has been updated and refined since this text was written. The updated method is described in an internal CADDETC document, copies of which will be available for review at the Washington meeting.

A standardised abstract test suite (ATS) for each Application Protocol shall be developed by the Application Protocol developers and approved by SC4 as a separate 300 series Part of ISO 10303. The ATS is the complete set of abstract test cases embodying all test purposes necessary to perform conformance testing of Application Protocol implementations. Abstract test cases are independent of implementations and are used to produce comparable results from the conformance testing of different implementations. An executable test case is derived from an abstract test case in a form which allows it to be run on the implementation under test. Requirements on the structure and content of the ATS are provided in ISO 10303-33. The abstract test suites are developed by the developers of the respective application protocol.

The process for creating the abstract test suite is:

1. Generate the appropriate set of Application Reference Model test purposes;
2. Generate the appropriate set of Application Interpreted Model test purposes;
3. Generate any other test purposes;
4. Combine test purposes into abstract test cases;

5 Document input specifications;

6. Document verdict criteria for each test purpose.

### 9.11.1. Application Reference Model test purposes

Each application object of an application protocol shall result in at least one Application Reference Model test purpose through the following rules:

- each application object results in a single test purpose stating the application object name;
- each required attribute of an application object results in a test purpose stating the application object name with the name of the attribute;
- each optional attribute results in two test purposes, one stating the application object name with the attribute present, one stating the application object name with the attribute not present.

Each application assertion of an application protocol documents the relationship between two application objects. The relationship is described as application object A to application object B and also as application object B to application object A. Each application assertion shall therefore result in at least two Application Reference Model test purposes describing each relationship through the following rules:

- each application assertion having a cardinality of zero results in a single test purposes stating the application object name A related to zero application object name B;
- each application assertion having a cardinality of one results in a single test purpose stating the application object name related to one application object name;
- each application assertion having a cardinality of more than one results in a single test purpose stating the application object name related to many application object name.

### 9.11.2. Application Interpreted Model test purposes

Every entity is associated with at least one Application Interpreted Model test purpose. Each optional attribute in an entity results in the definition of two test purposes, one with the attribute present and one with the attribute not present. Entities with simple type attributes shall require one test purpose for each attribute.

### 9.11.3. Other test purposes

Additional test purposes may be required because the application protocol references an external standard for use by implementations of that application protocol.

> EXAMPLE – ISO 10303-201 references ISO 3098-1 for use as its predefined text font.

Domain test purposes may be required to cover general requirements of the application domain or to constrain test purposes that need addressing in a particular or different contexts.

> EXAMPLE – ISO 10303-201 Annotation_subfigure allows scaling. In order to cover this general requirement, additional test purposes are required to address the different scaling 1:1, 1:n, n:1.

> EXAMPLE – ISO 10303-201 Chained_dimension_pair does not specify or constrain the dimensions which may be used. Test purposes covering sensible combinations are therefore required to address particular contexts.

### 9.11.4. Combine test purposes into abstract test cases

The quality of the abstract test cases relies on the expertise of the persons assigned to the task of combining test purposes to form small, realistic abstract test cases. The Application Activity Model may be a source for identifying suitable scenarios for the abstract test cases. Where this is used, the scenario shall be assigned an identifier and appropriate test purposes assigned to that abstract test case identifier. Care must be taken to ensure that test purposes allocated to an abstract test case are in the same conformance class.

### 9.11.5. Document input specifications

Each abstract test case has two input specifications, one for pre-processor conformance testing and one for post-processor conformance testing.

The pre-processor input specification shall describe the model that is to be created on the system under test in Application Reference Model terms. The specification shall be written clearly and unambiguously. Each test purpose assigned to that abstract test case shall be covered in the input specification. Application objects that are not covered in the test purposes assigned to that abstract test case may be used to complete the input specification.

> EXAMPLE – ISO 10303-201 requires a Drawing to be present in each input specification. Rather than create or document test purposes purely as an “administrative task” a drawing may be present in each input specification without an associated test purpose.

The post-processor input specification may mirror the pre-processor input specification. In this event, a suitable reference to the pre-processor input specification will suffice. Where the post-processor input specification does not mirror the pre-processor input specification a complete description must be given of the contents of the ISO 10303-21 physical file to be produced.

### 9.11.6. Document verdict criteria for each test purpose

Verdict criteria are statements of the requirements that must be satisfied for a “pass” verdict to be assigned during conformance testing. They are assertions on the observable output of an implementation under test.

Each verdict criteria shall be assigned the prefix “PRE-” for pre-processor testing and “POST-” for post-processor testing. Each verdict criteria shall be completed by a unique identifier relative to that test purpose e.g. an integer.

Each Application Reference Model test purpose shall have at least one verdict criteria for pre-processor testing and at least one verdict criteria for post-processor testing.

Application Reference Model test purposes that relate to application objects shall have their pre-processor verdict criteria documented in terms of the Application Interpreted Model element which is expected to be output in the ISO 10303-21 physical file. Post-processor verdict criteria for Application Reference Model test purposes shall require that the application object is present after the ISO 10303-21 physical file has been read in to the CAx system.

Application Reference Model test purposes that relate to application assertions shall have their pre-processor verdict criteria reference the Application Protocol mapping table to ensure validity. The corresponding post processor verdict criteria shall require that the relationship is present.

Additional verdict criteria for Application Reference Model test purposes may be required to check that required data values are present and correct.

Each Application Interpreted Model test purpose shall have at least one verdict criteria for pre-processor testing and may have one or more verdict criteria for post-processor testing[16]. The mandatory pre-processor verdict criteria requires that the Application Interpreted Model element specified in the test purpose is present in the ISO 10303-21 physical file.

# 10. Integrated resource development

Assigned to: WFD

## 10.1. Methods overview

Assigned to: WFD

creation phase (modular schema creation, cover generic vs. application resources)

extension phase

## 10.2. Integrated resource development procedures

Assigned to: JPF

(table) same structure as 10.2

JPF: as with the corresponding section on APs, a more formal approach may be useful here. Adam Polly, as Resource Integration project leader within WG4, may be able to offer support/insight to this section.

The procedures for the development of ISO 10303 Integrated Resource development differ according to the characteristics and scale of the development. Two types of Integrated Resource development are recognised:

- **creation** of new Integrated Resources; this is the development form used to put the initial Integrated Resources into place, or to add major capabilities to the Integrated Resources in response to novel industry and/or technological requirements.

- **extension** of existing Integrated Resources: this is the development form used to enhance or extend the Integrated Resources incrementally as additional requirements are identified during the Interpretation process for ISO 10303 Application Protocols.

Tables x and y summarise the procedures of Integrated Resource development used in each case.

JPF: these are incomplete.

[16] There may be a mandatory post-processor verdict criteria that requires that the corresponding ARM application object is present?

Table x: procedures for Integrated Resource development (creation phase)

| | Procedure | Type | Method(s) | Role(s) | Architectural element(s) | Documentation element(s) |
|---|---|---|---|---|---|---|
| 1. | A requirement for new Integrated Resource is proposed (Draft Resource Model) | | data modelling | Resource model development team (discipline experts) | Draft resource model | Draft resource part |
| 2. | Requirements analysed against existing Integrated Resources | | resource integration | Integration team | GPDM Existing resource models | |
| 3. | If requirement is confirmed, a New Work Item Proposal (to create a new Integrated Resource part) is initiated, or an extension to an “in work” Integrated Resource part is developed. | | | | | |
| 4. | A new Integrated Resource is developed to meet the requirements stated in the Draft Resource model | | Resource integration | Integration team, Resource part project | Integrated resource | |
| 5. | The Integrated Resource schema (or schemas) are reviewed and validated | | | Resource part project Other discipline experts | | |
| … | Qualification, Editing, Approval (as for APs in clause 10.2) | | | | | |

Table y: procedures for Integrated Resource development (extension phase)

| | Procedure | Type | Method(s) | Role(s) | Architectural element(s) | Documentation element(s) |
|---|---|---|---|---|---|---|
| 1. | A requirement for new Integrated Resource is identified during Application Protocol interpretation | | | | | |
| 2. | TO BE COMPLETED | | | | | |

## 10.3. Integrated Resource development methods

| |
|---|
| Assigned to: WFD<br><br>single generic model integration<br><br>shall not include application context dependent entities<br><br>Text taken from: Danner 9.2 N106 Integration process |

Resource integration involves four activities:

- the analysis of subject-area semantics as specified in draft resource models;
- harmonisation of resource constructs to principles of consistency;
- structuring of constructs to provide canonical forms;
- interfacing of constructs to provide a coherent set of resources.

### 10.3.1. Semantic analysis

Draft resource models are analysed to determine underlying meaning. Constructs are evaluated in terms of conceptual uniqueness and functional adequacy.

#### 10.3.1.1. Conceptual uniqueness

Each identified construct must be conceptually unique. Constructs are compared with those already present in the integrated resources to detect redundancies and conflicts.

Two types of conflicts are identified; naming conflicts and structural conflicts. Naming conflicts involve either homonyms where the same name is used for different concepts or synonyms where different names are used for the same concept. Structural conflicts are of four kinds:

t**ype**: a type conflict is where the same conflict is represented using different modelling elements.

**dependency**: a dependency conflict is where the same relationship between entities has different cardinalities.

**definitional**: a definitional conflict is where different mandatory attributes exist for the same concept.

**behavioural**: behavioural conflicts are where different rules regarding data integrity are established.

#### 10.3.1.2. Functional adequacy

Each of the resulting constructs in the integrated resources is traceable to an element of established overall goals and scope of an information domain in the data specification architecture. Every construct is established to fulfil a particular requirement. Examples and test cases are employed to evaluate the utility of the construct in terms of its declared purpose.

| |
|---|
| SPL: what kind of test cases here? |

Associations among resource constructs are identified for possible interfacing in a later stage of the integration process.

Ambiguities are removed by combining and dividing concepts, removing optional attributes, and coordinating attribute dependencies.

### 10.3.2. Harmonising

Harmonising provides non-redundancy of constructs, resolution of identified conflicts, and consistency of modelling. Constructs are conceptually aligned by resolution of detected conflicts. Techniques used to resolve conflicts include the creation of a generalised construct to accommodate conflicting requirements, creation of a specialisation of a construct, and removal of constructs for later resolution when sufficient knowledge is available for confident resolution.

#### 10.3.2.1. Modelling consistency

Of particular importance for modelling consistency is the identification of existence dependency among related concepts. Existence dependency involves a relationship between concepts where an instance of one concept is incomplete (i.e., cannot be present) without the presence of an instance of another. The modelling approach taken in ISO 10303 is to have the dependent concept reference the concept upon which it is dependent. This modelling principle is consistent with modularisation such that the phased development of the standard involves extensions that build upon fundamentals in earlier releases.

#### 10.3.2.2. Conceptual nature of construct

Integrated resource constructs are generic and conceptual in nature to provide shareability among multiple product types and application domains and implementability within a diverse heterogeneous environment of computer platforms. The integration process ensures that constructs are conceptual in nature. Constructs in the integrated resources convey semantics that logically describes product data concepts. These constructs do not include ideas or mechanisms that are motivated by convenience in practices, computer technologies, or efficiency requirements for implementation.

#### 10.3.2.3. Placement in the integration architecture

Constructs in the integrated resources are placed logically in the data specification architecture. Changes to either the constructs or the architecture may be made during the integration process when incompatibilities are identified. Over time, because of the impact on the already integrated constructs, it is easier to consider changes in a proposed additional construct than in the data specification architecture, although adding constructs rather than changing the architecture may result in requirements not being met.

The result of the harmonising process is a minimal set of consistent constructs that are ready for structuring.

### 10.3.3. Structuring

Constructs are structurally aligned by remodelling the concepts consistent with both the identified semantics of a draft resource model and the modelling conventions and canonical forms of the integrated resources. It is within this activity that voids are detected and resolved. Structuring provides completeness, structural consistency and structural precision with respect to semantic intent.

#### 10.3.3.1. Entity, relationship, and attribute specification

The specification of entities, relationships, and attributes is the primary integration issue. Entities and their attributes are specified consistent with the scope of the integrated resources. Relationships are specified such that references are based upon existence dependency (i.e. dependent entities reference entities upon which they are dependent). Common canonical structures are employed for consistency and efficiency. Entities, relationships, and attributes are added to resolve voids for completeness of constructs.

#### 10.3.3.2. Generalization

Concepts are modified to be appropriate for the scope and context of the integrated resources (i.e., free of any specific product data application context). Concepts that elaborate upon the semantics of more generic concepts that have been previously integrated are specified as specializations.

#### 10.3.3.3. Subject-area constraints

Constraints are specified during structuring. Constraints include relationship constraints and attribute-value constraints. Relationship constraints establish subject-area rules for references between concepts. Attribute-value constraints establish subject-rules for data consistency and integrity.

#### 10.3.3.4. Modularization

Modularization divides the requirements documented in a draft resource model among a number of manageable and conceptually consistent constructs.

#### 10.3.3.5. Integrated resource constraints

Constraints that establish rules specific to the scope and context of the integrated resources are incorporated at each intermediate phase of integration. Such constraints typically cross module boundaries (e.g. rules that ensure consistency between definition and representation data). These constraints are often implemented as additional entities, relationships, attributes, and rules.

### 10.3.4. Interfacing

Constructs are conceptually and structurally related to other constructs within the integrated resources. Interfacing maintains modularization of constructs with a minimum of inter-construct references.

References between constructs are controlled to ensure consistency and manageability of the specialization. References between concepts that are in different constructs (e.g. schemas) are controlled by the data specification architecture. Consistent with the architecture, the more specialized constructs reference the more general constructs. Existence dependency rules determine reference directions between constructs. The controlled references minimize the impact of change and thus minimize upward compatibility issues.

### 10.3.5. Integration conventions

Templates are used to capture semantics that involve directed binary association entities and composition entities. They are also used in the specification of management resources.

| |
|---|
| Danner 10.1 N106 Interpretation strategy<br><br>Danner 10.2 N106 Interpretation process |

# 11. Application protocol implementation

| |
|---|
| Assigned to: YY<br><br>Editorial instructions: include data exchange, data sharing, database implementation, computer application interoperability ("co-operative use of APs").<br><br>Previously implementation architecture. State the requirements on implementations that derive from the architecture e.g., "an implementation shall be based on the aim". This clause will cover four aspects of implementation: |

## 11.1. Data exchange

## 11.2. Data sharing

| |
|---|
| Initial text for this section produced by Steve Brett (CADDETC)<br><br>NOTE: This was documented as being out of scope in Greenville minutes - approved SC4 NWI however puts this back in scope. |

### 11.2.1. Requirements for data sharing

The introduction to ISO 10303-1 (and every other part of ISO 10303) states that the nature of the ISO 10303 description of product data "...makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases...". ISO 10303-1 goes on to describe a ISO 10303 database implementation as one in which the internal schema of the database conforms to the schema in an Application Protocol and additional implementation-specific requirements in an Application Protocol. It does not say what is meant by "sharing product databases".

It is generally accepted that there is a significant industrial requirement for something called "data sharing" that is different from "data exchange". In order to assess the nature of support for these in ISO 10303, it is necessary to first establish exactly what they mean and how they are different.

ISO 10303-1 defines data exchange as "the storing, accessing transferring and archiving of data.". It does not define data sharing. Note that this definition is independent of any implementation method: in particular, it is not restricted to file exchange. In fact, this definition includes any process that establishes access to data.

It is often the case that "data sharing" is associated with the concept of a shareable database, and is really being compared with "neutral file exchange" rather than "data exchange" as defined above. It is difficult to separate-out the implementation-specific features in this comparison.

For the purposes of this discussion, it is useful to define a "ISO 10303 Data exchange" as "the transfer of ISO 10303 data instances conforming to a single application protocol" [9]. However, it should be noted that there is an open issue regarding the cardinality of the relationship between a ISO 10303 data exchange and an Application Protocol. A useful data exchange should have the following features:

- a meaningful set of data is transferred;
- the meaning is preserved throughout the transfer;
- the transfer is in an identified context and for an identified purpose.

High quality data exchange requires an exchange specification that is highly specific and tightly constrained. Each system needs to know exactly what types of data are being transferred and what constraints apply to data values and integrity.

Note that the specification/model of a shareable data base may require very different characteristics, such as being very flexible and specifying as few constraints as possible.

So, what is meant by “data sharing”? In its broadest sense, it can be understood to include two different requirements: instance sharing and access management.

Maybe this should be integrity management and access management? We are really talking about rules that are applied for each transaction (though application may sometimes be delayed).

Data sharing: Instance sharing

Those industries that have to manage multiple inter-related sets of data for objects over a period of time require that, from the user perspective, one thing is represented by one instance, regardless of the number of applications that are used to access that instance. That is, when one thing is represented in a computer system, a single instance or set of instances should represent that thing. An example of a thing and its representation is a ship’s engine and the instance that is a surrogate for that engine in the ship’s maintenance system. Another example is the shape of the engine and the set of B-rep geometry data that is a representation of that shape.

Note that wireframe, surface, b-rep and CSG geometries are not simply different representations of the same thing: each model captures a slightly different information set: that is, each model has a different meaning. Thus, the CSG model has information about the procedural construction of the solid model in a CSG CAD system. There is no equivalent meaning in the b-rep model

If multiple instances are created for the purposes of data security or access efficiency, then the instances should be managed such that the end-user is not necessarily aware of that multiplicity. This is an implementation feature: from the viewpoint of the conceptual model, it is not necessary to consider more than one instance, and further discussion will use the term “single instance” on the understanding that this means a logically single instance.

Instance sharing can be summarised as the use by applications of logically common data instances (potentially concurrently).

Access management

Many organisations need multiple (potentially concurrent) access to data from different applications and/or different organisations. These organisations have a requirement for access management, which may be applied in several forms and on several levels. All of the following may need to be managed during data access:

- security: who is allowed access to the data?
- approval: what uses of the data are approved (different uses may be allowed for different people or classes of people)?
- concurrency: access by multiple parties or applications to the same data in a manner that does not create anomalies in the data;
- configuration control: modification of some data may have consequences for other data (for example, changing the shape of a component may mean that the shape of the part that houses that component needs to be reviewed);

- business-specific processes (e.g. a company may require that specified people are notified when some data is changed).

Note that the so-called instance sharing requirement could be seen as another access management requirement. It is simply at a lower level: concerned with enforcing structure and constraints associated with the underlying nature of the data rather than business rules or applications of the data (i.e. related to the conceptual model?).

### 11.2.2. ISO 10303 support for data sharing

Support for data sharing breaks down into two categories: provision of the required functionality by applications and support for that functionality in the data that is manipulated. Thus ISO 10303 cannot support the sharing of security classification data if that data is not specified in a ISO 10303 Application Protocol. Of course, if this was missing, it would not mean that ISO 10303 prevented the sharing of that data: it just would not support it.

The required functionality for data sharing will always be provided by applications, but can be supported by ISO 10303 through the specification of conceptual models/ISO 10303-22 events for that functionality. Some levels of rule checking (e.g. integrity checks) will typically be provided by a DBMS. Others could be provided by an ISO 10303-22 interface and the combination of DBMS and ISO 10303-22 layer should provide much (if not all?) of the functionality required.

It is generally agreed that ISO 10303, as designed, supports testable data exchange using application protocols, although there may still be some issues to resolve about the testability and how this is achieved [x.1].

Note that, when a shareable database is accessed (e.g. via ISO 10303-22), then each transaction can be considered to be an individual data exchange, and should meet all the requirements for data exchange.

There is currently no consensus on the extent to which ISO 10303 supports instance sharing. It is agreed that ISO 10303 is unlikely to prevent instance sharing, but the extent to which it will support it is a matter of debate.

It is possible to imagine several scenarios that are supported (at least to some extent) by ISO 10303, that qualify as instance sharing.

- Two types of application (A and B) share some common data requirements and an Application Protocol is written to specify data transfer between those applications. Irrespective of whether transfers are through ISO 10303 physical files or ISO 10303-22 interfaces, instance sharing can be achieved by transferring data between the applications using ISO 10303. In fact it is clear here that instance sharing can be achieved using neutral file exchange.

- A database is considered to be a third type of application (C), and an Application Protocol is written to specify data transfer from A to C and another Application Protocol for B to C. Where data types are common to both Application Protocols, they are part of one or more Application Interpreted Constructs. Thus, an instance created in a type A system can be stored in a type C via the Application Protocol A-C. The same instance can then be accessed from a type B system using Application Protocol B-C. While this is feasible with ISO 10303, the standard says nothing about the data structure of application type C, which may need to be changed/extended every time a new type of application needs to be supported.

## 11.3. Database implementations

## 11.4. Interoperability

# Annexes

## A. Bibliography

Assigned to: JPF/SPL


[1] Atlanta report of Ad Hoc Committee on Implementation Schemas - Phil Kennicott

[2] See ISO 10303-1.

[3] ISO 10303-41

[4] NIST GPDM paper

[5] Federal Information Processing Standard Publication 183, Integration Definition for Function Modelling (IDEF0), FIPS PUB 183, National Institute of Standards and Technology, December 1993,

[6] Kirkley, James, III, and Seitz, Brian K. "STEP Framework - Concepts and Principles", Version 3.000, July 17, 1991. ISO TC184/SC4/WG5 N6.

[7] Palmer, Mark. "AP Framework", 14 March 1993.

[8] PDES, Inc. "Testing Criteria Requirements Analysis Project, Stage I (TC-RAP I), Final Report", May 22, 1989.

[9] WG5 N101 P1: Workshop on ISO 10303 development methods, August 11, 1994. Julian Fowler, Steve Brett.


## B. Relationships to other architectures

assigned to WG10

- ANSI/SPARC
- OSI
- EDIFACT
- OMG/CORBA
- POSC
- EDIF ....

### B.1 Relationship of ISO 10303 to other architectures

NOTE: Previously one section now split to relationship of step to architectures AND methodologies

WFD (originally approx. 10 pages - now 5 pages?)

JPF 6/6/95: Should this be assigned to WG10?

## B.2 Review of ISO 10303 modelling concepts with respect to the ANSI/SPARC Three-Layer Architecture

### B.2.1 ANSI/SPARC three-layer architecture

The ANSI/SPARC three-layer architecture was first published in 1975 by the ANSI/X3/SPARC Study Group on Database Management Systems [WG5/N105 p20[14]]. This model was proposed as the core of a framework for DBM systems, but has since been widely used as a basis for other IT system architectures.

The architecture specifies three types of model, namely external, conceptual and internal (see figure B-1). Each type of model has a specific use and a defined relationship with the other types. The three types of model (corresponding to three layers) work together within the architecture to support the concept of data that is accessible from multiple applications or business perspectives and that may be stored in and moved between multiple implementation forms.

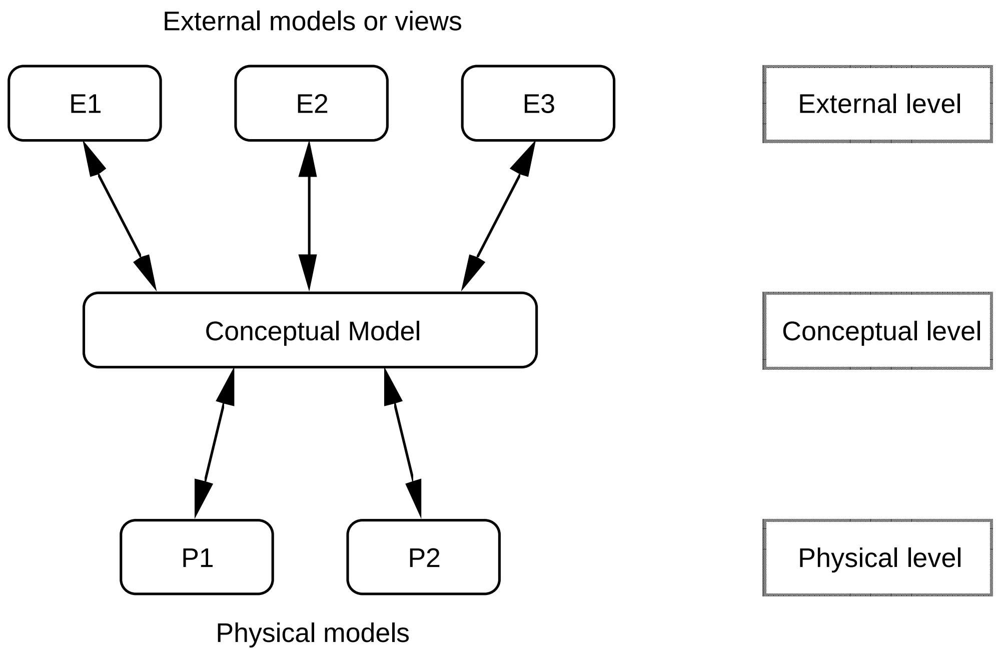


Figure B-1: ANSI/SPARC three-layer architecture

The conceptual layer contains a single model (within a given context) that is the basis for integration of the data used by different applications or stored in different formats. Models in this layer need to be stable over a long time scale, and so must not embody details that are specific to any application or any storage format, as these details are liable to change over short time scales.

The external layer contains one or more external models or “views”. Each of these is specific to a given application or business view and maps to a subset of the conceptual model in such a way that the data described in the external model can be held in the format of the conceptual model. Typically, an external model may contain subtypes of entities in the conceptual model and may constrain permitted data values through rules.

The internal or “physical” layer contains one or more physical models. A physical model is a complete specification for a data structure which is implemented through statements written in an application-specific Data Definition Language. The data structure must conform to the physical model in the sense that an exact, reversible mapping between them is defined. Each physical model is optimised for a specific implementation scenario (e.g. a relational database for archiving or a working form for computer-intensive processing). Any data structure that conforms to an internal model must be able to accept data that conforms to the associated conceptual model.

### B.2.2 The ISO 10303 Architecture

The ISO 10303 architecture is described in detail in chapter 6. The main elements of this architecture are summarised here and analysed in terms of the ANSI/SPARC three-layer architecture.

The ISO 10303 physical file format specifies the syntax of and rules for the construction of a ISO 10303 exchange file. The physical file format does not of itself constitute a complete physical model: rather it is the combination of this format with the Application Interpreted Model of a specified Application Protocol that constitutes the physical or “implementation” model for that Application Protocol. The physical model for an exchange file is currently the only one specified by ISO 10303. When the ISO 10303-22 is finalised, this will combine with an Application Interpreted Model to specify the physical model for transaction-based access to ISO 10303-compatible data.

The Application Reference Model of an Application Protocol is a formal description of the data required to support an identified business process. The Application Reference Model is written in application and industry-specific terms and takes no account of any requirements for the data to be used in another context. The Application Reference Model is an external model.

The Application Interpreted Model of an Application Protocol is derived from the Application Reference Model and the appropriate Integrated Resources. The Application Interpreted Model defines a data structure that is capable of capturing the full semantics described in the Application Reference Model. However, the Application Interpreted Model may be partitioned differently from the Application Reference Model and will typically embody additional constraints. The Application Interpreted Model is not simply the Application Reference Model re-stated in terms of Integrated Resource constructs: it is designed to support a business process in a manner that allows the data to be available and useful for other applications/processes, and so cannot be classified as an external model.

In all but one respect, the Application Interpreted Model qualifies as a conceptual model: it is designed to allow use of data by multiple applications and is independent of any implementation requirements. However, the scope of the Application Interpreted Model is limited to the scope of the Application Reference Model and in this respect, any single Application Interpreted Model is highly specific to an application or business view.

Because of the way they are designed, where the Application Interpreted Models of ISO 10303 Application Protocols have overlapping scopes, the overlap constitutes a portion of the information model that is common to the Application Protocols involved. That is, the union of two Application Interpreted Models with overlapping scopes would be a single model that covers the scope of both Application Interpreted Models. Note that such a union is not constructed within the current methodology. {ISSUE 1} Thus, it is the notional union of all Application Interpreted Models that constitutes the conceptual model in ISO 10303. However, even this interpretation has problems, as the Application Interpreted Models embody all of the application or business-specific constraints that are supposed to reside in the external models.

The Integrated Resources are a minimal set of non-redundant models that together form a single integrated model. All application/business-specific and implementation-specific details are excluded and the model is designed to support the data requirements of multiple applications. The Integrated Resources have all the characteristics of a conceptual model, but they are not used as a conceptual model {ISSUE 2} inasmuch as they are not implemented. The role of the Integrated Resources is to be the basis for developing Application Interpreted Models. This role is outside the scope of the ANSI/SPARC three-layer architecture.

SJB: Note that it may be useful to give a comparison between STEP and the Federated Database Systems architecture also.

ISSUES

1. This is ensured by a complex procedural approach. It is the extent to which this is guaranteed to be true that is questioned by some. Also: is it practical to expect that the current procedure will still be able to maintain this “single model” as the scope grows. Finally: what will the quality of this model be like, as it is developed by integrating each addition as the need occurs, rather than by populating a pre-determined framework - in this respect, the union-of-Application Interpreted Models may not qualify as a conceptual model, as the ANSI/SPARC architecture is built around the idea of a pre-existing conceptual model that is closer to the Integrated Resources.

2. This is arguable. Each Application Interpreted Model can be seen as an external model in relation to a conceptual model that is the Integrated Resources, but there are more problems with this view (further discussion needed here).

3. One thing that is clearly adopted from the ANSI/SPARC architecture is the principle of separating physical and conceptual aspects into different models. However, this principle is violated by making the STEP physical file a direct implementation of the Application Interpreted Model.

4. The ARM can be considered to be a conceptual model that integrates all of the external views of different CAD systems of similar functionality.

# C. Relationship to other methodologies

JPF: is this assigned to WG10?

# D. Technical discussions

JPF 6/11/95: this annex is used to collect material with an historical focus, or that is informative background to the rest of the document. As and when this annex grows, more structure may be needed.

## D.1 Informative Annex for 7.3

When PDES, Inc. first began it's activities, a principle effort of the program was the testing and validation of the evolving STEP standard. At the time (1988-89), the STEP standard was a monolithic document composed of EXPRESS schemas. These schemas were organised according to the type of data structures they contained: all the geometry was in one schema, tolerances in another schema, and features in yet another schema. The organisation of the standard was product-data-centric. Like IGES, it was simply a semantically "flat" specification of data structures to be used to exchange product data.

This presented a challenge for the Testing and Validation Team. How do you test "geometry" or "tolerances"? IGES was "tested" in the field; after years of use, improvements to IGES were made based on real-world usage of the standard. Since field testing is expensive and undesirable, there was strong feeling in the STEP community and within PDES, Inc. that there was great value in testing the new standard before it actually became a standard.

Therefore, PDES, Inc. undertook a study to establish an approach for testing and validation of the product-data-centric models in STEP. The study was called the Testing Criteria Requirements Analysis Project. A significant finding of this study is as follows:

> "Based on the Analysis of the Current Environment, many of the improvement opportunities indicated the need of data models that can specifically satisfy a given application context.

> Testing of these models under the same application context will yield validation of the usefulness of these models. If these application context models are "composed" by using the PDES, Inc. baseline models[17] as their resources, the validation of these application context models also validates the data constructs and definitions of the baseline models for the application context."

Based on this finding, the project recommended the development of a Context-Driven Integrated Model (CDIM) for testing the STEP data models.:

> "A context driven integrated model is an integrated model which is composed by data definitions and data construct relationships, constraints, etc., from the PDES, Inc. Baseline models, that can provide data requirements for a specific application context. Since the structure of this integrated model is driven by the requirements within an application context, it is a context-driven integrated model (CDIM). CDIMs will be the bridge between application user's requirements and the conceptual models in the PDES, Inc. testing environment."

Thus, the need to test the standard required the definition of an application context within which to conduct the testing.

For several years prior to this PDES, Inc. study, the IGES community had been working on IGES Application Subsets. These subsets, developed in response to IGES "flavours", documented a subset of the IGES entity set and specified rules for using the entities correctly within a specific application domain. The term “subset” was replaced by the word “protocol” because the specification was not simply a subset of the IGES standard, but a protocol for using the standard.

While the reasons that led to their development differed, the concepts inherent in Application Subsets and CDIMs were sufficiently alike to give birth to a new element of STEP - the Application Protocol.

### D.1.1 Application Context Taxonomies

The notion of a context for the use of product data is easy to understand because people recognise the use and role of context in everyday conversation. Describing the context, however, is much more difficult. What exactly are the salient features of the "circumstances in which a particular event occurs" and how can they be described?

Since the introduction of the AP concept, there have been several attempts to define taxonomies, ontologies, and classification structures for the description, analysis, and planning of the scopes of Application Protocols. Brief descriptions of relevant attempts are as follows:

#### D.1.1.1. From "AP Framework" (Palmer) [2]

The purpose of this document is to:

> "Define and implement necessary improvements to the policies, process, and tools for planning and managing the development, deployment, and maintenance of cost-effective STEP APs."

One tool is an AP Classification system that makes use of the following designations:

<u>AP Scope Classifiers:</u>

[17] Since the data models which constituted STEP were in a constant state of flux, PDES, Inc. found it necessary to establish "baseline" models for testing.

- **Primary classifiers**: for top-down planning of AP projects and collaboration between enterprises and standardisation committees

  A product domain

  B life cycle tasks (functional processes) supported

  C product structure

- **Secondary classifiers**: for detailed definition of scopes

  D types of product data (UoFs and AICs)

  E views/disciplines supported, e.g., Industrial Robotics, Structural Engineering

  F data usage (implementation form+)"

### D.1.1.2. From "STEP Framework - Concepts and Principles" (Kirkley and Seitz)

This work is a wide-ranging document touching on many aspect of STEP development. The relevant sections concerning Application Context include:

"The STEP Framework currently identifies four domains (dimensions) of STEP:

- Generalisation/specialisation
  - GPDM level
  - resource model level
  - application protocol level
- Lifecycle
  - requirements analysis (concept)
  - design
  - prototype
  - test
  - implementation
  - operation
  - maintenance
- Category
  - function
    - presentation
    - tolerance

      - geometry & topology
      - material
      - form features
      - mechanical products
      - manufacturing technology
      - product structure and configuration management"
    - discipline
      - architecture
      - electrical products
      - mechanical products
      - ship structures
      - sheet metal
      - software"
  - Version
    - time-dependent relationships between models

### D.1.1.3. From "Testing Criteria Requirements Analysis Project" (PDES, Inc.)

The TC-RAP final report present the most detailed taxonomy, but it biased to Aerospace manufacturing. A partial list of the categories follows:

- Product Data Applications
  - Engineering
    - Conceptual Design
    - Detail design/drafting
    - Lofting
    - Analysis
    - Design Verification Testing
  - Manufacturing
    - Manufacturing Engineering
    - Mfg Control
    - Production

    - Inspection
        - …
    - Facilities
        - …
    - Logistics
        - …
    - Manage Product
        - …
    - Government
        - …
- PDES Use Applications (use PD application for context)
    - Exchange
    - Integration
    - Automation
    - Drawingless
- Other Applications
    - Engineering Disciplines
        - Mechanical Products
        - Composites
        - Electrical Products
        - AEC
        - Tech. Manuals
    - CAD/CAM/CAE/CIM System Capabilities
        - 2D wireframe & Drafting
        - 3D wireframe & Drafting
        - BREP Solids
        - CSG Solids
        - FEM systems
        - Simulation systems

- NC Systems
- CAPP Systems
- CMM Programming & analysis systems
- WIP/Config. Mgt./MRP Systems

wcbMay 9, 1995: perhaps these should all be moved to an annex.

### D.1.2 Use of taxonomies

The most common thread through the extant taxonomies, frameworks, and classifications is the Lifecycle categorisation. This is followed by categorisation based on the empirical typing (kinds) of product data; and by then by industry sectors or product type.

Taxonomies have not been widely used in the planning and development of STEP APs for a number of reasons. The first and most obvious is that the classification schemes are not well-structured; if Lifecycle, Product Data type, and product type were treated as axes, they would not be orthogonal to one another.

A further, and perhaps more cogent, reason is that most uses of product data cannot easily be pigeonholed in a single "cell" from such a classification structure. Most AP developers see the use of the AP spread across many cells; the result is that even with a (supposedly) well-defined scope, there is constant scope creep.

# E. Known issues and constraints

This annex covers two aspects of the ISO 10303 Architecture and Methodology. The first deals with “Frequently Asked Questions” (FAQs), i.e., statements of and responses to issues related to the ISO 10303 architecture and methodology that are common raised in ISO 10303 development activities.

The second clause within this annex covers issues against the ISO 10303 architecture and methodology that have arisen during the development of this part, or that of other parts to ISO 10303, and for which there is no current, accepted resolution.

NOTE – issues against this document, as opposed to the architecture and methodology, are recorded in Annex E.

## E.1 Frequently Asked Questions

JPF 6/5/95: if this annex is to be useful, it must contain a comprehensive list of the FAQs perceived by members of WGs 4, 6, and 5/10. The source of many of the FAQs – WG3 – needs to be involved as well: action for WG3 convener. To be effective, each FAQ needs to be structured as follows:

- Title
- Statement of the issue/question
- Summary of the response to the issue/question
- Reference to the clause(s) of this document and/or other documents where more details relevant to the response may be found.

JPF's "Introduction to the STEP Architecture and Methodology" presentation may be a good starting point to collect/solicit the basis for this section.

## E.2 Outstanding issues

Issues (both here and in Annex E) need to follow a standard format, as used in tracing and responding to CD and DIS ballot issues.

| | |
|---|---|
| Issue number: | allocated by ISO 10303-13 project team |
| Date: | date issue received by ISO 10303-13 project team |
| Issue title: | assigned by Issue Owner (or by the ISO 10303-13 project team, if not provided) |
| Issue owner: | the person(s) or organisation responsible for raising the issue |
| Classification: | categorisation of the issue, assigned by the issue owner. Suggested values are "major technical", "minor technical" and "editorial". |
| Status: | assigned by the ISO 10303-13 project team. Possible values are "open", "resolved" (a resolution has been identified, agreed, and implemented), "rejected" (the ISO 10303-13 disagrees with the premises of the issue as stated), "unpersuasive" (the ISO 10303-13 project team accepts the issue as valid, but has agreed that no change to the document is necessary), "accepted" (the issue has been accepted as valid and as requiring change to the document; the change has not, however, been agreed or implemented), "redundant" (issue rendered irrelevant by other changes), or "unclear" (a statement of clarification is requested from the issue owner). |
| Description: | supplied by the issue owner |
| Proposed solution(s): | supplied by the issue owner; these may be supplemented by the ISO 10303-13 project team during the issue resolution process. |
| Resolution date: | date on which the issue was declared "resolved", "rejected", "unpersuasive", or "redundant". |
| Resolution: | description of the changes resulting from resolution of the issue; in most cases, this will take the form of instructions to the editor. |
| Commentary: | justification of the decision taken by the ISO 10303-13 project team in response to the issue. |

By tracking all issues raised during the development, consensus and approval stages, the ISO 10303-13 project team will seek to maintain full change control and traceability over the document.

JPF: the following is given as included in the post-Sydney version of this document (WG5 N139). These do not, however, appear to be issues!

1. The following key principles underlie the ISO 10303 Architecture and Methodology:

i. An architecture for product data, providing stability and extendibility[18].

ii. Traceability of data to industry needs.

iii. Standardisation of industry application semantics, in as much detail as can be agreed.

2. ISO 10303 is developed through the interaction and co-operation of groups with specific industry or application expertise, and those with a “vision” across the scope of ISO 10303/SC4.

## F. Issues Log

| |
|---|
| Assigned to: JPF/SPL<br><br>JPF: still to be put into the format described in Annex D above. |

The following issues/comments were raised at the Sydney meeting, March 1995

ISSUE - SYD/1: Is the traceability of data to industry need related to the Application Activity Model?, the Application Reference Model?, the Application Interpreted Model? or combinations of these?

ISSUE - SYD/2: Does the specification of industry need for data (instance) over-constrain the usage of an Application Protocol, either in its applicability to scopes other than that for which it is designed (e.g., applying AP203 to maintenance data), or by preventing further constraints on the context?

ISSUE - SYD/3: The principle of traceability of data to industry need is vague, and may in fact be derived from the principle of standardisation of industry application semantics. Is the requirement for traceability of data to industry needs, or for mappability of industry needs to data?

ISSUE - SYD/4: Does the structure of the ISO 10303 Integrated Resources prevent the representation of “non-product” geometry, such as the shape of the environment in which a building is to be constructed?

ISSUE - SYD/5: Does the emphasis on <u>product</u> data in the terminology of ISO 10303, and the naming of the **product** entity, cause confusion with respect to understanding the scope of applicability of ISO 10303?

ISSUE - SYD/6: Are Application Reference Models and Application Interpreted Models at the same level of abstraction? If they are, are both needed in an Application Protocol, or is the Application Reference Model a development tool that should not be included in the final documentation of the Application Protocol as a standard?

ISSUE - SYD/7 (from Thomas Thurman, March 21 1995) The interpretation methodology is flawed in that there are multiple levels of information discovery that occur during the development of an Application Reference Model, but the current methodology documentation ignores that fact. For instance, once an Integrated Resource is available that is almost purely representation (e.g., geometry), an Application Protocol project should be able to identify that with a simple reference to the kinds (Application Interpreted Constructs) of geometry needed. An elaborate model in the Application Reference Model of detailed information requirements is redundant and most likely will contain errors of fact that will:

- need to be corrected;

[18] The DEC paper states that "... the purpose of an architecture is to accommodate change". The Shell data modelling methodology includes amongst its requirements the need for stability in the face of changing needs.

- mislead the Interpretation project.

## Part 13 comments - Washington

Comments which have been addressed in N22 are marked with an asterisk (*)
Comments which have been partly addressed in N22 are marked with two asterisks (**)
Comments which may have been addressed through other comments are marked with a question mark (?)

| | | |
|---|---|---|
| 1 | Sanford | Include the following definitions/concepts:<br>• data exchange requires a contract between exchange partners which defines at the meta level the elements to be exchanged and a mechanism for measuring success<br>• an exchange standard is a public exchange contract allowing a wide range of partners to readily participate<br>Therefore, the problem is to define what needs to be exchanged and the meta-model for it |
| 2 | Sanford | Either update figure 1 or include an additional figure – see “STEP on a page” diagram |
| 3* | Danner | add a sub-clause on Mapping Tables to clause 6 |
| 4* | Danner | change the title of 6.6 to “Model Specifications” |
| 5** | Danner | make 7.4 a separate clause (12) (within a separate section?) covering assumptions and requirements from data specifications (e.g., implementation schemas), “conceptual instances”, implementation forms |
| 6* | Danner | “established” is mis-spelt in the first paragraph of the Foreword |
| 7* | Danner | Reword paragraph 1 of Scope: “This description provides … methodology. It also includes …”. |
| 8* | Danner | issue on definition of Application Protocol: should also include AAM, ARM, and mapping table [issue on Part 1] |
| 9* | Danner | Abbreviations: IAS = Industry Application Scope |
| 10* | Danner | Abbreviations: remove reference to PMAG |
| 11* | Danner | Include in 4.2:<br>• multiple views (projections) of a product over the entire life-cycle<br>• aggregations of characteristics with multiple representations<br>• application context is functionally determined |
| 12* | Danner | page 9, last line: delete “below” |
| 13 | Danner | page 10, figure 1: update to include mapping table |
| 14* | Danner | 4.3, second paragraph after figure 1: “The key element … architecture depend is …” |
| 15* | Fowler | make the paragraph following figure 1 a NOTE |
| 16* | Danner | page 11, under product property representation: delete spurious “of” |
| 17* | Danner | footnote 6: “… thought or activity …” |
| 18 | Danner | page 15, second paragraph: second rationale (first is traceability of data to product) |
| 19 | Danner | same paragraph: existence dependency does not of itself provide stability |
| 20* | Danner | 5.3.3, point 1: this aspect of the method provides for modularity |
| 21* | Danner | 5.3.3: Semantic and Syntactic Rules paper is by Danner, Sanford, and Yang |
| 22* | Danner | 5.3.5, last paragraph on page 17: make the two types of usage constraint explicit by using a bullet list |
| 23 | Danner | 5.3.6, last paragraph: explain how AICs are “… not intended to identify all shared IR constructs …” |
| 24* | Danner | 6.2, third line: change “data objects” to “application objects” |
| 25 | Danner | 6.2/6.3: introduce the idea that the ARM specifies a domain ontology |
| 26 | Danner | 6.3: ensure consistent use of AIM as a term: in the AP document, the AIM consists of the short form EXPRESS plus the mapping table |
| 27 | Danner | 6.4: introduce that idea that the IRs are an abstract cognitive model |
| 28 | Danner | 6.7: see issue 26 above |

| | | |
|---|---|---|
| 29* | Danner | 6.8: introduce the abbreviation ATS at the same time as the expanded term |
| 30* | Danner | Include mapping tables in the editorial instructions for 7.1 |
| 31 | Danner | 7.2/7.3: IRs are not (just) a vocabulary; rather, they are a vocabulary and a grammar that together constitute an abstract cognitive model |
| 32* | Danner | 7.3, first line: delete "act" |
| 33 | Danner | 7.3, first paragraph: ARM is an application-specific domain ontology |
| 34* | Danner | 7.3.2, Industrial Application Context: this is specified by the entire AAM, i.e., it includes the "out of scope" activities and flows |
| 35* | Danner | 7.3.2, Industrial Application Scope: selected elements of AAM for AP |
| 36* | Danner | 7.3.3: delete references to specific attempted taxonomies for APs |
| 37 | Danner | 7.3.5.1: must be described consistent with standard data elements (data architecture). IRs ≡ underlying semantics (abstract cognitive model). Any application view be can represented using the abstract cognitive model. |
| 38* | Danner | 7.3.4.1: remove reference to PDES, Inc. |
| 39 | Danner | 7.3.4.2: NO! the IRs are not a vocabulary (see above). Also, "abstract" and "fuzzy" are not synonyms. |
| 40 | Danner | 7.3.5.3: this section is written more like a "white paper" than a Reference Manual |
| 41* | Danner | 7.3.5.4: first line, change "accurate" to "accurately". |
| 42 | Danner | Context = real world circumstances in which something is done<br>Domain = processes, knowledge and "agents" that produce something in a context |
| 43* | Danner | 7.4.2.1, third bullet on page 31: background knowledge is necessary for correct inferences (use of data). |
| 44 | Danner | 7.4.2.1: use agreed terminology to distinguish between data communication and data integration |
| 45* | West | 11.3.2.3: adding constructs rather than changing the architecture may result in requirements not being met. |
| 46** | West | General comment: a lot of repetition |
| 47 | West | Introduction, purposes of Part 13: the rationale generally requires an historical perspective (especially for something pragmatic) |
| 48 | West | Scope, first bullet: so more than just APs |
| 49** | West | Executive Summary. Comes first. Most executives won't get this far! Almost by definition this must be free standing. It may repeat or summarise material elsewhere. Our rule is maximum of 1 page! |
| 50 | West | Clause 5: this clause covers purpose and requirements as well as objectives. Either the title or the content of the clause should change. |
| 51 | West | 5.1: is the provision of "… standard data specifications for unambiguous communication of information …" the primary objective of STEP, or more what has been achieved so far? |
| 52* | West | 5.1, second paragraph: change "demanded" to "lead to". It should be clear that the design principles (especially that related to context-dependent semantics) were choices that were made (or driven). |
| 53 | West | 5.1.4, first paragraph: NOT TRUE! |
| 54? | West | 5.1.4, second paragraph: "… a single integrated communication standard is not a viable solution" – can this be demonstrated (proved)? |
| 55 | West | 5.2, second paragraph on page 15: what sort of existence dependence? data dependence? real-world? Current practice does not support extensibility |
| 56 | West | 5.2, third paragraph on page 15: this is the biggest weakness of the current methodology |
| 57 | West | 5.3: some historical perspective is very helpful (essential even) in understanding a methodology that has evolved to meet emerging requirements. However, most of this is <u>not</u> historical – just a statement of what is. |
| 58* | West | 5.3.2, last line of first paragraph: delete spurious "the". |
| 59 | West | 5.3.2: what <u>is</u> a product (or what is not)? |

| | | |
|---|---|---|
| 60 | West | 5.3.2: is a product definition a view of a product, or a view of the definition of a product. Text says the latter; should be the former. |
| 61 | West | 5.3.6, first paragraph: there is a higher level missing. |
| 62* | West | 6, editorial instruction: what is GEDM? |
| 63 | West | 6.2, JPF comment: anything in the executive summary should draw from elsewhere. |
| 64 | West | 7.3: like most other sections this covers the whole architecture and methodology, from a perspective. |
| 65 | West | 7.3, second paragraph: "… innovative and unique …" – a big claim! |
| 66 | West | 7.3, third paragraph: are the information requirements given by clause 4.2 or the ARM? |
| 67 | West | 7.3.2, Industry Application Scope: this is a usage. |
| 68 | West | 7.3.3.1: this is not thought through or justified |
| 69 | Kennicott | General: the function of STEP tends to be described as "communication"; the original functions were communication (of physical files), database access (or programming interface), and archiving. All are important, and all should appear in the manual. NIPDE adopted the term "data exchange" to embrace all three. |
| 70 | Kennicott | Page 14: the term "deep structure" should be defined. |
| 71 | Kennicott | Page 16, 5.3.1: The Tokyo IPIM was intended as a place holder, and was never to be implemented. While the observations are correct from the standpoint of a person not realising this, they are unfair to the editors of the IPIM. |
| 72 | Kennicott | Page 19, 6.2: Reference should be made to the mapping table, particularly in view of its importance, as brought out at the (WG10) workshop (on AP Interoperability). |
| 73 | Kennicott | Page 24, 7.3.3: I question the value of a discussion of context taxonomies. It is unclear that they have had an effect on the standard. They rather appear to be only an artefact of our preparation process. |
| 74 | Kennicott | Page 77, 10.11: A question has been raised in the US as to whether this section accurately represents the WG6 consensus. |
| 75 | Wenzel | Abstract (cover page): the agreed purpose of the document is "Documentation of current Architecture and Methodologies". General suitability is neither agreed nor intended. |
| 76 | Wenzel | Foreword: what is the criteria for inclusion in the list of ISO 10303 parts? |
| 77 | Wenzel | Page iv, third boxed note: This (data sharing and archiving using APs) was the intention and an initial requirement! |
| 78 | Wenzel | Page vi, fourth boxed note: Disagree! |
| 79* | Wenzel | 5.1, first paragraph: change "… for unambiguous …" to "… to cover unambiguous …" |
| 80 | Wenzel | Footnote 5, page 13: Effectiveness is a qualitative property, efficiency is a quantitative one. If a solution is not effective, its efficiency is undefined! |
| 81 | Wenzel | 5.2, boxed note on page 14: this was an axiom of the current methodology/architecture. Don't touch! |
| 82 | Wenzel | 5.2, boxed notes on page 15. First note: not at all, but it is current practice.<br>Second note: Disagree! This model is to document where we are, not where we want to get to. |
| 83 | Wenzel | 6.2, boxed note: I'd prefer a different presentation over a repetition. |
| 84* | Wenzel | 7.3, boxed note no. 2: change "This resolves …" to "This is intended to resolve …". |
| 85 | Wenzel | 7.3.3, NOTE – irrelevant for the current status, exclude. |
| 86 | Wenzel | 7.3.5.3: exclude, irrelevant for the current situation |
| 87 | Wenzel | 7.3.5.4: exclude, not correct |
| 88 | Wenzel | 10.1.2, boxed note: document what we have, not what could be. |
| 89 | Wenzel | 10.6.2, mapping table example (table 2): syntax definition and explanation should be given. |
| 90 | Wenzel | Annex B: add federated databases |

| | | |
|---|---|---|
| 91 | Sanford | Redundancies in the document structure:<br>4. Executive summary with architecture and methodology overview<br>5. Objectives (actually design guidelines of data architecture and methods overview)<br>6. (Data) Architecture components<br>7. Data Architecture<br>More distinctly separate Architecture from Methods<br>9. Is the beginning of the Methods section<br>Add a higher layer under architecture, to include three points:<br>• customer focus, i.e., scope, AAM, leading to:<br>• data architecture, supporting:<br>• an implementation architecture |
| 92 | Sanford | Introduction: second paragraph is redundant with Part 1. [Issue with Supplementary Directives?] |
| 93 | Sanford | Introduction, list of purposes: is it appropriate to a standard that the reference manual should “be a basis for improvement …”? |
| 94 | Sanford | Introduction, last paragraph: state which chapters constitute the two sections – otherwise appears to be in conflict with 11 chapters which are the first level of decomposition in the table of contents |
| 95 | Sanford | Scope, fourth bullet: good, but where? |
| 96* | Sanford | Definition of AIC: “a logical grouping of <u>interpreted constructs</u> that …” |
| 97 | Sanford | Definition of AP: add “… and its relationship to industrial needs.” [Issue against Part 1]. |
| 98 | Sanford | IUT is not defined |
| 99 | Sanford | PICS is not defined |
| 100 | Sanford | Definition of conforming implementation: discuss certification? I.e., certified to meet instead of satisfies. [Issue against Part 31] |
| 101* | Sanford | Definitions of EXPRESS and EXPRESS-G: change to data specification language |
| 102* | Sanford | Definition of fail (verdict): remove – verdict is sufficient; this is common English in the context of verdict. [Issue against Part 31] |
| 103* | Sanford | Definition of inconclusive (verdict): remove – verdict is sufficient; this is common English in the context of verdict. [Issue against Part 31] |
| 104* | Sanford | Add a new definition.<br>Interpreted construct: the association of a resource construct with a specific need. It is the atomic element of an AIM or AIC, resulting from interpretation. |
| 105* | Sanford | Definition of ontology: change to “… classify a domain of discourse.” |
| 106* | Sanford | Definition of pass (verdict): remove – verdict is sufficient; this is common English in the context of verdict. [Issue against Part 31] |
| 107 | Sanford | Definitions of pre-processor and post processor: change ‘internal format of a particular computer system’ to ‘some other private format’. Even ‘private’ is questionable, e.g., IGES to STEP. [Issue against Part 31] |
| 108* | Sanford | Definition of token separator: remove – the byte count is superfluous (not part of token); the text is common English. |
| 109 | Sanford | Clause 4, introductory paragraph: is it appropriate to a standard that the document’s should be “… as the basis for future improvement”? |
| 110 | Sanford | Clause 4.1, first bullet point: needs the concept of “long term utility” of data |
| 111 | Sanford | Clause 4.1, third bullet point: note that data is not necessarily independent of the processes which create or consume it |
| 112 | Sanford | Clause 4.1, end of last sentence: used to read “publicly available binding” which implied simultaneously computable and accessible. Need to preserve this thought. |
| 113 | Sanford | Clause 4.2: add a sixth fundamental principle – “Ensure standard computable bindings exist” |
| 114* | Sanford | Clause 4.3, second paragraph after figure 1: add “… all elements of the architecture <u>are dependent</u> is …” |

| | | |
|---|---|---|
| 115* | Sanford | Same paragraph, last sentence: change "such a representation …" (to end of para.) to "to a specific purpose in a specific industrial application domain". |
| 116 | Sanford | Third paragraph after figure 1: omits AICs and the idea of binding resource constructs to their use. |
| 117 | Sanford | Figure 2: ensure that the part of the diagram labelled "DATA" is shown to be an example |
| 118 | Sanford | Clause 4.3, note at the bottom of page 11: isn't this the same as saying that STEP demonstrates the well accepted concept of referential integrity? |
| 119 | Sanford | Clause 4.4: implies only "others" have vision; suggests others have data modelling expertise and overall integration responsibility |
| 120 | Sanford | Clause 4.4: include a matrix of joint responsibilities (not WG based) |
| 121* | Sanford | Clause 5: Change to "<u>Design Principles</u> of the …" |
| 122** | Sanford | Detailed comments on clause 5: see marked up copy of earlier draft; these issues have to verified as still pertaining as relevant to the text for this clause as in N13 |
| 123* | WG10/P1, Arlington | Move the Executive Summary to the Introduction, under a separate sub-clause heading. |
| 124 | WG10/P1, Arlington | Add to clause 6: role of implementation forms (source: AP203 implementation schema discussions – requirements on all implementations; results of ad hoc committee, Atlanta?) |
| 125 | Sanford | Clause 5: restructure around the following design principles:<br>• human interpretable<br>• computer interpretable<br>• syntactic integration (single style)<br>• structural integration (single structure for sharing of data)<br>• semantic integration<br>• context-dependent semantics<br>• stability<br>• extensibility<br>• usability<br>• Producibility<br>• interoperability |
| 126 | Debbie Washington | Proposed FAQ: The concept of AICs and IRs seems to be the same. Where exactly do they differ, and as a developer how do I know which to use or research for possible overlaps of information? |
| 127 | Debbie Washington | Why do they keep changing the document guidelines? They are making it harder and harder with all the new rules and constraints. (I commented that at least there is some boilerplate areas now, which were not in place three years ago). |
| 128 | Debbie Washington | Once the ARM and AIM are complete, no one looks at the AAM. Why can't that section be dropped from the final version? |
| 129 | Debbie Washington | How are the EXPRESS long and short forms developed, and why is it necessary to include both in the AP? |
| 130 | Debbie Washington | Why is it called EXPRESS? There is nothing expedient about it! |

| | | |
|---|---|---|
| 131 | C. Rehling | As the document is intended to be a REFERENCE Manual, I would not mind if several issues are addressed in more than one place. In principle, I agree with your statement in the 3rd box on page 12, but if, e.g., aspects of AP Harmonisation were discussed in a section by themselves and mentioned where the structure of APs, in particular the ARM, is explained, I think that would be helpful. Cross-references should be given, though.<br>The thing is that many people looking something up in Part 13 may not be aware of how things interrelate. Thus, they will not automatically also check for AP Harmonisation if the think they need to look for AP development. Therefore, the larger set of knowledge has to be made available to them in a structured manner, and they need to be guided to adjacent subject areas. This is only possible if some issues are discussed in more than one place, and if proper cross-references exist. |
| 132 | C. Rehling | Scope of Part 13 (first two bullets in "1. Scope", page 1). The first and second bullet seem to contradict each other: either all standards within SC4, which covers 10303 and 13584 (and MANDATE?), or only 10303.<br>I think Part 13 would be of more use if it applied to all standards within SC4. Thus, ISO 13584 needs to be included and mentioned already in the introduction (page vi). If the methods of 13584 are not the same as those of 10303, the document may need to be split in 3 or 4 parts: The two that there are already, plus 1 for the architecture and 1 for the methodology of ISO 13584.<br>Given that a number of AP Project teams see the need for libraries in their models (which hopefully will lead to a defined way of using 13584 within 10303), it would be rather confusing if two separate Reference Manuals were to be developed. |
| 133 | C. Rehling | Definition of "representation" ("3. Definitions", pp.2 - 7)<br>The term "representation" is not defined. I think it should be defined or at least explained. In particular for people who are not native English speakers, a definition (or at least an explanation) would be helpful to develop the same concept (because the term translates into a number of words in our mother tongues, and we do not necessarily know which is closest to the intended usage of "representation" ). |
| 134* | C. Rehling | Remove PMAG from the list of abbreviations (p. 8).<br>As Part 13 is a reference manual and not a STEP History Guide, and as PMAG will soon cease to exist, any later users of Part 13 should not be forced to work through the history of the development of the organisation nor of this document. Thus, all references to PMAG should be removed. (Exception: "Comments to Reader" box on cover sheet for those who review Part 13 drafts). |
| 135 | C. Rehling | Editorial - Fig. 1 (p. 10): Explanation of dashed line arrow is not given. I also would like to suggest to frame the figures, and to clearly separate any explanations from the rest of the text. |
| 136* | C. Rehling | Editorial - Fig. 2 on page 35 should in fact be Fig. 3 (there is "another" Fig. 2 on page 11). |
| 137 | C. Rehling | Editorial - Is it possible to clearly indicate the section of the document (Architecture or Methodology) by changing the numbering? That may help in using the document when individual sections or clauses are quoted. Currently, one would have to know that "9" indicated the first chapter of section II.<br>Would numbers I.1 through I.8 for the Architecture and II.1 through II.3 for the Methodology be allowed under ISO style requirements? |

| | | |
|---|---|---|
| 138 | C. Rehling | Section 5.2, page 14; 4th paragraph, last sentence:<br>"Because Application Protocols are based on a single integrated model, applications that can read the data produced according to one Application Protocol are able to read data produced by any Application Protocol."<br>• I think the term "produced by an AP" is misleading. Suggestion: "Exchanged using an AP"<br>• I think the statement is not generally true, as APs may subtype IR constructs. In such a case, only the data contained in the "lowest common supertype" is understood by more than one AP. Depending on, e.g., the binding used in the exchange structure, some data are understood by both implementations (external binding) or no data are understood by both (internal binding).<br>• The last two lines on page 14 indicate that subtyping is allowed, contradicting (in my opinion) the sentence quoted above. |
| 139* | C. Rehling | Section 5.3.7, page 18; 1st para, 2nd sentence:<br>"Within each Application Protocol, the specification is further partitioned ..."<br>To my knowledge (and stated by Mary Mitchell in Greenville), APs are not required to have Conformance Classes. Thus, I suggest to change the above statement to read " ..., the specification may be further partitioned ..."<br>(This fact is also given in the NOTE in section 6.7, page 20) |
| 140 | C. Rehling | Section 7.3.4.1, page 25/26, third bullet:<br>"of the in-scope information, a subset OF INTEREST can be specified as the information requirements that must be met by the Application Protocol"<br>That effectively means that this OF INTEREST subset constitutes the lowest conformance classes of an AP, and it simultaneously indicates that there have to be at least two conformance classes (low = OF INTEREST, highest = all the AP) for this AP, doesn't it? If so, please state it, so that the concept of Conformance Classes is tied in to this AP development process. |
| 141 | C. Rehling | Section 5.3.6, page 18; 1st paragraph, 2nd and 3rd sentences<br>I think that the two levels on which relationships are said to exist between APs are not really two levels, i.e., not independent of each other.<br>If two particular APs that do not only by definition use the same set of IRs in addition also use the same subtypes of specific IR constructs, I think that does not add a new quality or dimension.<br>I would rather say that while all APs share a common foundation (IRs), some may also share a number of common "pillars". The AICs would be these "pillars". AICs are "more" than pure IR constructs. This "more" should be seen somewhere outside the AIM of those APs, i.e., if AICs are identified in the interpretation process, some ARM (and maybe AAM) level requirements must be resembling each other for those APs with common AICs.<br>What is the link between AICs and Units of Functionality or Functional Data Groups (AP 214 term)? Can UoFs or FDGs be the ARM-equivalents of AICs? |

142 C. Rehling

Concerning the draft Part 13, I would like to suggest to address the (AP) Harmonisation issue not only in different places where parts of the contents of the AP are discussed. but also as a separate chapter, i.e., "orthogonal" to the AP thread.

I think that some kind of harmonisation of the development of different APs is needed regardless of how it is performed. Thus I think the statement "There are pros and cons against Harmonisation activities." (section 7.3.5.3, page 28, 3rd paragraph) should be removed.

I think that AP Harmonisation consists of several layers.

Layer 1: Even before the AP is an active project in STEP, i.e., when the AP scope is drafted, other APs in the same domain or the same industry segment should be looked at. Possibly/Hopefully, a Core Model (like researched in AEC) can add value at this point.

Layer 2: While the AP is being developed, i.e., AAM and ARM are created, the process called "AP Harmonisation/Harmonising (section 11.3.2)" in your Part 13 may be applied.

Layer 3: In preparation of the AIM, AP interpretation takes over, and possible AICs are identified.

I think one has to be careful not to "over-harmonise" in the middle layer, because there is a trade-off between efforts involved there and the later interpretation stage. However, I would expect efforts in the field of harmonisation of AP scopes (layer 1) to pay off much sooner and also to add value to STEP by ensuring that APs fit together. (For example, if a suite of APs is to be developed supporting the design activities in an industry segment, it should be ensured that all design activities are captured either by the Predesign or by the Design AP, such that no "holes" in the activities in design remain that are not covered by either AP.)

Concerning the box on page 38 (section 10.1.2), I suggest that as long as there is no definitive feedback from different teams testing different approaches to ARM development, this part of the document be seen as a living document. Perhaps the entire issue of harmonisation could be made an annex as soon as the situation is not stable. Thus, it can be updated easier.

# Index